

MINT

version 1.0.1

Users' Manual

Anna Górska
Maciej Jasiński
Joanna Trylska



Biomolecular
Machines
Laboratory

MINT is a free software; you can redistribute it and/or modify it under the terms of the GNU General Public License version 2, as published by the Free Software Foundation.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.



Copyright (C) 2013 – 2014: University of Warsaw

Contents

1	Introduction	4
1.1	Citing MINT	4
2	Quick reference	5
2.1	What can MINT do?	5
2.2	What do you need to run MINT?	6
2.3	Definitions	6
3	Installation	6
3.1	Required external python packages	6
3.2	MINT	7
4	Running MINT	7
4.1	Analysing the DNA versus RNA structures	8
4.2	Parameters specified in the CONFIG.MINT configuration file	8
4.3	Example	10
5	Output files	11
5.1	description.txt	12
5.2	MINT.xls	12
5.2.1	Pairs	12
5.2.2	Pairing	13
5.2.3	Secondary structure motifs	13
5.2.4	Stacking	13
5.2.5	Pi interactions	15
5.2.6	Nucleotides characteristics	15
5.3	Timeseries	15
5.3.1	Motifs clusters	17
5.3.2	Average motifs	17
5.4	Dot bracket	18
5.5	Secondary structure drawings	18
5.6	Inputs for external visualization tools	18
6	Methods	19
6.1	Workflow	19
6.2	Loading the DNA or RNA conformations	20
6.3	Calculating physico-chemical features	20
6.3.1	Hydrogen bonding	20
6.3.2	Donors and acceptors	20
6.3.3	Modified nucleotides	22
6.3.4	Geometric isomerism of base pairs	23
6.3.5	Stacking	23
6.3.6	Ion-II interaction between a nucleobase and phosphate oxygen	25
6.3.7	Representation of RNA motifs	26

CONTENTS	3
6.3.8 Motif-search algorithm	30
7 Analysis of trajectories or many RNA/DNA conformations	31
7.1 Clustering	31
8 Visualization	34
8.1 Representing motifs on the tertiary structure	34
8.2 Representing hydrogen bonding and stacking on the tertiary structure	35
8.3 Representing secondary structure changes	41
8.4 Correlations	43
9 Appendices	44
.1 Adding hydrogens to a PDB structure	44
.2 Exemplary description file	44

1 Introduction

MINT (Motif Identifier for Nucleic acids Trajectory) is an automatic tool for reading and analyzing three-dimensional structures of nucleic acids, their molecular dynamics trajectories (written in CHARMM, Gromacs, NAMD, LAMMPS, or different formats) or other conformation sets.

MINT deals with both DNA and RNA molecules. However, since mainly RNAs acquire complicated 3D folds, we write this manual in the context of RNA. The analysis includes hydrogen bonding, stacking and secondary-structural motifs. For many conformations of the same nucleic acid molecule MINT provides various physicochemical properties as a function of simulation time (or frame), as well as statistics providing an overall-view of the changes in these properties occurring in the inputted conformation set (e.g., trajectory).

Please direct your comments and questions about the MINT package to:

agorska@cent.uw.edu.pl or m.jasinski@cent.uw.edu.pl

Centre of New Technologies, University of Warsaw

Banacha 2c, 02-097 Warsaw, Poland

phone: +48 22 5543 683

The web interface to analyze the PDB files containing single RNA or DNA structure and the source code of the MINT package to analyze multiple conformation sets (for example, molecular dynamics trajectories) is available at:

<http://mint.cent.uw.edu.pl>

or can be accessed via:

<https://bionano.cent.uw.edu.pl/Software>

MINT is distributed under the terms of the GNU Public License. A copy of the GPL is provided with the distribution and is also available at <http://www.gnu.org>.

1.1 Citing MINT

If you find this software useful please cite: Anna Górska, Maciej Jasiński, Joanna Trylska, *MINT: software to identify motifs and short-range interactions in trajectories of nucleic acids*. Nucleic Acids Res. 2015 May 29. doi: 10.1093/nar/gkv559 PubMed PMID: 26024667.

2 Quick reference

2.1 What can MINT do?

For a single PDB file of an RNA/DNA molecule MINT provides:

- regions forming helices,
- regions forming loops, bulges, interior loops, junctions,
- regions forming pseudoknots,
- nucleotides creating triplets,
- all Watson-Crick edge base pairs,
- all non-Watson-Crick edge base pairs,
- all stacking and anion- π interacting nucleotides,
- the number of Watson-Crick edge hydrogen bonds per nucleotide,
- the number of non-Watson Crick edge hydrogen bonds per nucleotide,
- the stacking energy – Van der Waals and electrostatic energies and their sum per nucleotide,
- visualizations of the computed parameters.

For trajectory (or set of conformations) of the RNA/DNA molecule (e.g., derived from molecular dynamics trajectory) MINT provides:

- residues forming helices, triplets, pseudoknots and other motifs,
- all base-pairs: Watson-Crick edge and non-Watson-Crick edge, along with the conformation,
- clusters of secondary structure motifs and average motifs along with hydrogen bonding (in Watson-Crick edge and non-Watson-Crick edge contacts),
- all stacking, anion- π interacting nucleotides,

All of the detected structures/motifs is described also with its occurrence in the trajectory: percentage of frames and their specific numbers. How often given structure was present in the trajectory is a measurement of its stability.

MINT computes also various global parameters for a given trajectory:

- the average number of Watson-Crick and non-Watson Crick edge hydrogen bonds per nucleotide,
- the average stacking energy – Van der Waals and electrostatic energies, their sum per nucleotide,
- average secondary structure,
- visualizations of the above listed parameters (see section 8)

2.2 What do you need to run MINT?

- Python2.7 and external python packages: `numpy`, `BioPython`, `MDAnalysis`, `xlwt` and `pympler`. For details see section 3.1.
- a PDB file with a full-atom RNA structure (including hydrogens).
- a trajectory file (e.g. in the .dcd format) to use the `Traj` mode.
- a version 1.6+ of the JAVA Virtual Machine to run VARNA 1 the visualiztion applet that MINT uses to generate secondary structure pictures.

2.3 Definitions

Here we outline a few concepts/definitions used in this manual and software.

- Secondary contacts – both nucleotides interact with the Watson-Crick edge (WC), see Figure 5.
- Tertiary contacts – at least one of the nucleotides interacts with its non-Watson-Crick edge (non-WC), see Figure 5.
- RNA secondary structure – a structure created by the WC base pairs, excluding pseudoknots, as in the standard dot-bracket notation.
- RNA tertiary structure – a structure created only by non-WC pairs.
- Motif – a loop, a bulge or junction; a helix is treated as a separate entity.

3 Installation

3.1 Required external python packages

Despite the python 2.7 (<http://python.org>), MINT uses also some external packages:

- `numpy` – the package with tools to manipulate multi-dimensional arrays (<http://www.scipy.org/Download>).
- `BioPython` – the main package for loading and managing the PDB structures (http://biopython.org/wiki/Main_Page).
- `MDAnalysis` [17] – used for reading the trajectory (<http://code.google.com/p/mdanalysis/wiki/Install>).
- `xlwt` – the package for writing .xls files. It is used by `csvToxls.py` script to write all output .csv files into the .xls files (<https://pypi.python.org/pypi/xlwt>).
- `pympler` – the package determining the memory usage of the objects in the python script (<http://pythonhosted.org/Pympler>).

- if you wish to use CORRELATIONS.py script (see paragraph 8.4), you should install `matplotlib` python package: <http://matplotlib.org>.

All of the required packages have their home websites (provided in brackets above) where they can be downloaded from along with instructions. All of them can also be installed through `easy_install`, that is provided by the `setup_tools` package (https://pythonhosted.org/setuptools/easy_install.html#installing-easy-install):

```
easy_install numpy
easy_install Biopython
easy_install MDAnalysis
easy_install xlwt
easy_install pympler
```

If you encounter problems with installing `python2.7` or required packages, you can use the `Multipy` package. `Multipy` enables to set up a local virtual python environment and run any `python` script. You can download `multipy` from: (<http://code.google.com/p/multipy/> or <https://github.com/akheron/multipy>).

First, to install `python 2.7` in the `multipy` environment, type:

```
multipy install 2.7
```

then you can access the `python 2.7` `multipy` environment by using:

```
bash
. $(multipy activate 2.7)
```

and second, install the required packages inside the environment, using `easy_install` as in the example above.

3.2 MINT

Download and unpack the `MINT.tar.gz` package from the webpage:
<http://mint.cent.uw.edu.pl/index.php?strona=MintDownload>

4 Running MINT

Go to the `MINT` directory and type:

```
python MINT.py CONFIG_traj.MINT
```

and the program will perform a short run for the `example.pdb` and `example.dcd`, present in the `MINT/example` folder (see Section 4.3), with all parameters set to their default values. Results will be placed in the `MINT/example-traj` folder. If any error appears check whether you are using `python 2.7` and have all the required packages installed (see Section 3.1).

`MINT` uses a simple text configuration file, which specifies input parameters. Each line in the configuration file consists of a keyword identifying the parameter and its value. Comments are denoted by the `#` character. The syntax is:

```
keyword:option # comment
```

4.1 Analysing the DNA versus RNA structures

MINT includes necessary parameters for analysing both DNA and RNA molecules. By default MINT distinguishes the DNA molecules by the names of the residues in the input PDB file, it expects the DNA nucleotides to be named: Adenine: DA, Guanine: DG, Cytosine: DC, Thymine: DT. Other nucleotides will be treated as RNA or derivatives. However, with the option `all_nucs_DNA: 1` it's possible to force MINT to treat all standard nucleotides as DNA.

4.2 Parameters specified in the CONFIG.MINT configuration file

- **Mode** – [Single/Traj/Download] determines whether a trajectory or single structure will be analyzed. In the Single mode, only the conformation from the PDB file `file_name` parameter will be analyzed. In the Traj mode conformations from the `file_dcd` file will be analyzed.
- **out_name** – [directory and/or file name] specified file name will be the prefix for all of the output files. You can also add a relative path to the directory in which you want outputs to be created (e.g. specifying `./foo/bar` you will place your output files with `bar_` prefix in the `./foo` directory).
- **file_name** – [file name] input PDB file containing nucleic acid structure. The file is required in both Single or Traj modes.
- **file_dcd** – [file name] input file containing multiple conformations, trajectory for the Traj mode. The supported trajectory formats are `.dcd .xyz .trr .crd` as in the `MDAnalysis` package.
- **chains_names** – [list] the list of chain ids to be analyzed. If it is empty, the analysis will be performed for all chains. **Note, that for analysis all provided chains are treated as a single chain.**
- **first_frame** – [int] the number of the first frame to be analyzed from the trajectory file. **Note, that numbering starts from 0.**
- **last_frame** – [int/-1] the number of the last frame to be analyzed in the trajectory, if set to -1 the program replaces it with the number of the last frame.
- **stride** – [int] the number specifies how many frames to skip between two analyzed frames. When set to 0 the whole trajectory will be analyzed.
- **cutoff** – [\AA] the cutoff for the distance measured between the C1' carbons of every nucleotide. For nucleotides located in distance larger than `cutoff` the program does not search for hydrogen bonds. Default is 20.
- **cutoff_stacking** – [\AA] the cutoff for the distance measured between the centers of mass of every nucleobase. For nucleotides placed in distance larger than `cutoff_stacking` the program does not search for the stacking interaction. Default is 10.5.

- **vdw_cutoff_stacking** – [kcal/mol] the maximal value of the Van der Waals (VDW) energy for the stacking interaction. If the VDW energy between two nucleobases is smaller than the given **vdw_cutoff_stacking** parameter, the stacking interaction is detected. Default value: -0.5.
- **OP_stacking_distance_cutoff** – [\AA] the maximal distance between a backbone phosphate group and a nucleobase center of mass for the calculation of the anion- π interaction. If measured distance is smaller than given **OP_stacking_distance_cutoff** parameter the anion- π interaction is detected. Default is 4.5.
- **h_bond_atom** – [donor/hydrogen] indicates whether the hydrogen bond distance should be computed between a **donor** and acceptor or a **hydrogen** and acceptor.
- **h_bond_l** – [\AA] the maximal length of the hydrogen bond. Default value of 4 was set with **h_bond_atom:donor**.
- **h_bond_angle** – [degrees] the minimal angle of the hydrogen bond. Default is 140.
- **vmd** – [0/1] the binary parameter working only in **SingleOrTraj:Single** mode. If set to 1 a VMD application will be opened and the input structure coloured by secondary structures. This is possible only if VMD is properly installed and added to PATH (alternatively, it can be run from command line, see Section 8.1).
- **table_nucleotides** – [file name] the .csv file determining the hydrogen donors, acceptors and nucleotide edges for every nucleotide. By editing this file, one can remove certain interactions from the analysis or define new types of interactions. Default file **nucleotides_modified.csv** is stored in the MINT/data folder.
- **table_charges** – [file name] the .csv file listing the partial charges, Van der Waals radii and depths of the Lennard-Jones potential for atoms in nucleotides. These parameters are given for two force fields: AMBER [24] and CHARMM [9,16], but there is also a column MY_OWN for the user to put other parameters if needed. Default file **./data/charges_and_VDW_modified.csv** is stored in the MINT/data/ folder.
- **list_of_modified_nucs** – the two-column text file with the list of modified RNA nucleotides present in the RNA structures deposited in the PDB database (as of July 2014). Each row contains the residue name used for the modified nucleotide and a single-letter code (a, g, c, t, u) for the natural nucleotide from which the modification originates. Default **data/unknown_modified.fa** is stored in the MINT/data folder.
- **force_field** – [AMBER/CHARMM/MY_OWN] the name of the force field to be used by the program while computing stacking energies.
- **margin** – [0.0 – 1.0] the minimal portion of nucleotides that have to be common for both motifs to belong to the same cluster. Default is 0.8.
- **time_cutoff** – [0.0 – 1.0] the minimal portion of the conformations (part of the frames) from the trajectory where the motifs must be present in order to be incorporated into the cluster analysis. Default is 0.02.

- **max_memory_GB** – [GB] the maximal memory the single thread is allowed to use at one time. Default is 1.5.
- **threads** – [int] the number of CPUs to be used while analyzing the trajectory. Default is 1.
- **only_analysis** – [1/0] if 1, the program instead of running the whole analysis, reads the previously performed analyses, performs computations only for the missing frames and creates output files. You can turn on this parameter if your computations were disturbed for some reason. In this mode MINT uses only one CPU. Default is 0.
- **create_csvs** – [1/0] if 1, MINT will create separate csv file for each sheet present in the **_MINT.xls** output file. Only nonempty sheets and files are made. When analyzing very long trajectories MINT may produce csv files, even if the parameter is set to 0. This is related to the limitation of the .xls file format. In such case there will be a reference in the .xls file to see the .csv files.
- **pairing_cutoff** – [0.0 – 1.0] the minimal portion of the conformations (part of the frames) from the trajectory where pair must be present in order to be incorporated into the **pairing** table. Default is 0.1.
- **all_nucs_DNA** – [1/0] if 1, MINT will treat all standard nucleotides as DNA instead od RNA. Default is 0.
- **write_nucs_timeseries** – [1/0] if 1, MINT will save hydrogen bonds and stacking interaction analysis vs time in csv files. Requires installation of additional python library: **pandas**. Default is 0.

4.3 Example

The **example** directory of MINT contains the following inputs:

- **example.pdb** – a .pdb file with the atomic structure of the RNA molecule. One will benefit from the program mostly by analyzing RNA structures with complex secondary and tertiary structures.
- **example.dcd** – a trajectory file containing ten frames from molecular dynamics simulations performed with NAMD [19] and using the CHARMM force field.

In the **example-traj** following **outputs** are present:

- Structure description (see section 5 for details):
 - example_MINT_description.txt
 - example_MINT.xls
 - example_nucleotides_characteristics.csv
 - example_pairs.csv
 - example_pairing.csv

- example_helices.csv
 - example_motifs.csv
 - example_average_motifs.csv
 - example_motifs_clusters.csv
 - example_stacking.csv
 - example_pi_interactions.csv
 - example_dot_bracket.txt
- Visualization, secondary structure movie (see section 8 for details):
 - example_RNAStructML.xml
 - Visualization, PDB:
 - example_Hbonds-WC.pdb
 - example_Hbonds-non-WC.pdb
 - example_Hbonds-total.pdb
 - example_Stacking-VDW.pdb
 - example_Stacking-Coulomb.pdb
 - example_Stacking-total.pdb
 - Visualization, pictures:
 - example_Hbonds-WC.png
 - example_Hbonds-non-WC.png
 - example_Hbonds-total.png
 - example_Stacking-VDW.png
 - example_Stacking-Coulomb.png
 - example_Stacking-total.png

All the examples provided in this manual are derived from above files.

5 Output files

MINT generates multiple files both in the single frame analysis mode and in the trajectory mode. Names of the generated files begin with the `out_name` parameter value. The output files describe nucleotides in the way that they can be unambiguously identified. We use the representation containing a chain name, a residue name and its number. For example: the `N|GUA:521` represents the residue number 521 from the chain N, with the name GUA. The values posted in the `chain name|residue name:residue number` code are derived from the input PDB file.

5.1 description.txt

The `_description.txt` file contains a complete description of the structure for every analyzed frame. The exemplary description is shown in the end of this manual. The single frame description contains a complete list of used parameters, helices, motifs, triplets, pseudoknots. One can also find a list of WC, non-WC pairs, stacking and π interactions and their exact parameters, as well as a dot-bracket representation of the secondary structure. The frames are separated by the headers: `frame number`.

5.2 MINT.xls

The `_MINT.xls` file is a complete .xls (spreadsheet type) file collecting all calculated properties in separated sheets. If the `create_csvs` parameter is set to 1 a .csv file will be created for every present sheet. Only non-empty files/sheets are produced. In the first sheet of the `_MINT.xls` there is a legend explaining used sheet names. In the `Trajectory` mode of analysis the following columns are present in selected output sheets/files:

- `% of frames` – contain percentage of frames when a motif/pair/interaction was present, describes in how many frames of the trajectory given structure (e.g. a pair) was present. It is a measure of the stability of the interaction.
- `frame numbers` – contain frame numbers in which a motif/pair/interaction was present. The notation $2 \rightarrow 15\ 20 \rightarrow 26$ denotes that the structure can be found in frames number 2,3,4.. up to 15, and than frames number 20 to 26.

Below there is a description of the sheets forming the `_MINT.xls` file:

5.2.1 Pairs

The `pairs` sheet is a table containing all nucleobase pairs that appeared during the trajectory. The sheet contains the following columns: `1st nucleotide number`, `nucleotides`, `pair type` classified according to the interacting edges (e.g. WC-WC), `pair configuration (cis or trans)`, `vmd`, `% of frames`, `frame numbers`, the exemplary data record is shown below:

1st nucleotide number	nucleotides	pair type	pair configuration	vmd	% of frames	frames numbers
515	N G:515/ N C:536	WC/WC	Cis	chain N and resid 515 536	100%	0 → 9
516	N U:516/ N C:519	WC/Hoogsteen	Trans	chain N and resid 516 533	50%	3, 6 → 8
522	N C:522/ N G:527	WC/WC	Cis	chain N and resid 522 527	100%	0 → 9

Table 1: Example of the `pairs` table.

5.2.2 Pairing

The pairing sheet is a table enlisting the hydrogen bonding partners for every nucleotide. The `pairing_cutoff` parameter specifies the minimal part of frames in which pair has to be present in order to be taken into account. The partners are enlisted vertically and than coded: number of hydrogen bonds - nucleotide (chain|name:number) - number of frames. See the example below: The C nucleotide number 501 from chain N was forming 2 hydrogen bonds with G number 544 in

nucleotide number	nucleotide	interaction	interaction
500	N G:500	3-N C:545-5	
501	N C:501	2-N G:544-3	3-N G:544-2
502	N A:502	2-N U:543-1	3-N U:543-4

Table 2: Example of the pairing table.

3 frames and 3 hydrogen bonds with the same partner in 2 frames.

5.2.3 Secondary structure motifs

There are separate sheets created for different types of the secondary structure motifs:

- `helices`,
- `pseudoknots`,
- `triplets`,
- `motifs`.

Each of them contains the following columns:

- `nucleotides` - nucleotides forming the motif,
- `vmd` - ready to paste into the VMD residual identifiers,
- `% of frames`,
- `frame numbers`.

The `motifs` sheet contains also a `motif_topology` column – for explanation see 6.3.7. Below there is an example of the motifs table:

5.2.4 Stacking

The stacking sheet is table containing all the nucleotide pairs recognized as stacked. The table contains the following columns: `1st nucleotide number`, `nucleotides`, Coulomb avg, Coulomb `sd` (the Coulomb interaction energy in kcal/mol followed by its standard deviation), `VDW avg`, `VDW sd` (the VDW interaction energy in kcal/mol followed by its standard deviation), `Stacking-total avg`, `Stacking-total sd` (sum of the VDW and Coulomb interactions for the recognized pair in

motif topology	nucleotides	VMD	% of frames	frame numbers
7-5	N G:515-N C:536 ... N U:516-N G:515	chain N and resid 515 536 .. 516 515	100%	0 → 9
4	N C:522-N G:527 ... N A:523-N C:522	chain N and resid 522 527 .. 523 522	100%	0 → 9
0-6	N C:504-N G:541 ... N G:505-N C:504	chain N and resid 504 541 .. 505 504	100%	0 → 9

Table 3: Example of the `motifs` table.

kcal/mol with the standard deviation value), vmd, % of frames and frame numbers. In case of analyze of the single structure, columns with the standard deviations values will not be present. Example of the stacking sheet is presented below.

1st nucleotide number	nucleotides	Coulomb avg	Coulomb sd	VDW avg	VDW sd	Stacking-total avg	Stacking-total sd	VMD	% of frames	frame numbers
509	N A:509/ N A:510	25.64	2.25	-5.52	0.61	20.11	2.33	chain N and resid 509 510	100.00%	0->9
501	N C:501/ N A:502	6.23	1.80	-2.65	0.55	3.58	1.88	chain N and resid 501 502	100.00%	0->9
503	N C:503/ N G:544	1.54	0.03	-0.51	0.01	1.04	0.03	chain N and resid 503 544	20.00%	1 5

Table 4: Example of the `stacking` table.

5.2.5 Pi interactions

The `pi_interactions` sheet describing ion- π interactions is constructed analogously to the `stacking` table. With one additional column, `distance` containing the distance between the ion and the center of the mass of the second nucleotide.

5.2.6 Nucleotides characteristics

The `nucleotide_characteristics` sheet is a table enlisting all nucleotides and their description. This list is useful while evaluating how much a nucleotide is stabilized by its environment by both the hydrogen bonds and stacking interactions. Every nucleotide is described with the total number of hydrogen bonds (`Hbonds-total` column) broken down into WC pairs (`Hbonds-WC` column) and non WC pairs (`Hbonds-non-WC` column), and the stacking energy (`Stacking-total` column) broken down into two energetic terms: VDW (`Stacking-VDW` column) and Coulomb (`Stacking-Coulomb` column). Stacking energy, presented in *kcal/mol*, is originally computed for pairs. The value for a single nucleotide is a sum of all interactions of a given kind satisfying the `cutoff_stacking` and `vdw_cutoff_stacking` parameters. In the case of the trajectory analyze, columns described above contains average numbers and are followed by the columns with the values of standard deviation. Below there is an example of the `nucleotide_characteristics` table.

5.3 Timeseries

Not present by default (see `write_nucs_timeseries` parameter in configuration file). 6 CSV files containing the data that is used to construct the `nucleotide_characteristics` table described above: `Hbonds-WC`, `Hbonds-non-WC`, `Hbonds-total`, `Stacking-VDW`, `Stacking-Coulomb` and `Stacking-total`. In each file, every column represent data for a single nucleotide.

nucleotide number	nucleotide	Hbonds-WC avg	Hbonds-WC sd	Hbonds-nonWC avg	Hbonds-nonWC sd	Hbonds-total avg	Hbonds-total sd	Stacking-Coulomb avg	Stacking-Coulomb sd	Stacking-VDW avg	Stacking-VDW sd	Stacking-total avg	Stacking-total sd	VMD
502	N ADE:502	2.00	0.00	0.00	0.00	2.00	0.00	11.45	2.47	-14.00	0.99	-2.55	2.66	chain N and resid 502
515	N GUA:515	3.20	0.40	0.00	0.00	3.20	0.40	1.95	1.76	-17.10	1.26	-15.15	2.16	chain N and resid 515
517	N GUA:517	0.00	0.00	0.80	0.40	0.80	0.40	6.97	1.91	-10.22	0.54	-3.25	1.98	chain N and resid 517

Table 5: Example of the `nucleotide_characteristics` table.

5.3.1 Motifs clusters

The `motifs_clusters` sheet is a table enlisting all recognized motifs assigned to clusters (see Section 7.1 for details). Clusters are numbered starting from zero. The sheet is created only in the **Trajectory** mode. Below there is an example of the `motifs_clusters` table:

cluster number	motif topology	nucleotides	VMD	% of frames	frames numbers
cluster 0	4-0	A520-A533-...- G521-A520-	chain N and resid 520 533 ... 521 520	30%	3 → 4 9
cluster 0	7-5	G515-...-U516-G515-	chain N and resid 515 536 ... 516 515	70%	0 → 2 5 → 8
cluster 0	2-4	G515-...-U516-G515-	chain N and resid 515 536 ... 516 515	30%	3 → 4 9
cluster 1	4	C522-G527-...-A523-C522-	chain N and resid 522 527... 523 522	100%	0 → 9
cluster 2	0-6	C504-G541-...-G505-C504-	chain N and resid 504 541 ... 505 504	100%	0 → 9

Table 6: Example of the `motifs_clusters` table.

5.3.2 Average motifs

The `average_motifs` sheet is a table showing the average motif for each identified cluster. All listed average motifs are described with the vectors of the average numbers of hydrogen bonds in WC (WC) and non-WC hydrogen bonds (non-WC). The `average_motifs` sheet is created only in the **Trajectory** mode. Below there is an example of the table:

cluster number	motif topology	nucleotides	VMD	% of frames	frames numbers
1	4	C522-G527-...-A523-C522-	chain N and resid 522 ... 522	100%	0 → 9
	WC	3.0 3.0 2.8 ... 0.0 3.0			
	non-WC	0.0 3.0 0.8 ... 2.1 0.0			
2	0-6	C504-G541-...-G505-C504-	chain N and resid 504 ... 504	100%	0 → 9
	WC	2.9 2.9 ... 2.8 2.9			
	non-WC	0.0 0.0 ... 0.0 0.0			

Table 7: Example of the `average_motifs` table.

5.4 Dot bracket

The `dot_bracket.txt` file is a simple text file created only in the `Trajectory` mode. Sequential lines contain the dot-bracket representation of the secondary structure for corresponding frames of the trajectory. This file enables analyzing the evolution of the secondary structure in external tools. Below there is a fragment of the exemplary `dot_bracket.txt` file.

```
(((((.....((((((....)))).....)))))))  
(((((.....((((((....))....))))))))  
(((((.....((((((....)).....))))))))
```

5.5 Secondary structure drawings

Secondary structures coloured by the hydrogen bonding and stacking interactions (see Figure 1). Figures are generated using the `VARNA` program [7]. The nucleotide sequence is retrieved directly from an input PDB file. The secondary structure is computed from the list of WC base pairs – in the case of the single frame from an input PDB file and in the case of the trajectory from the list of the most represented WC pairs.

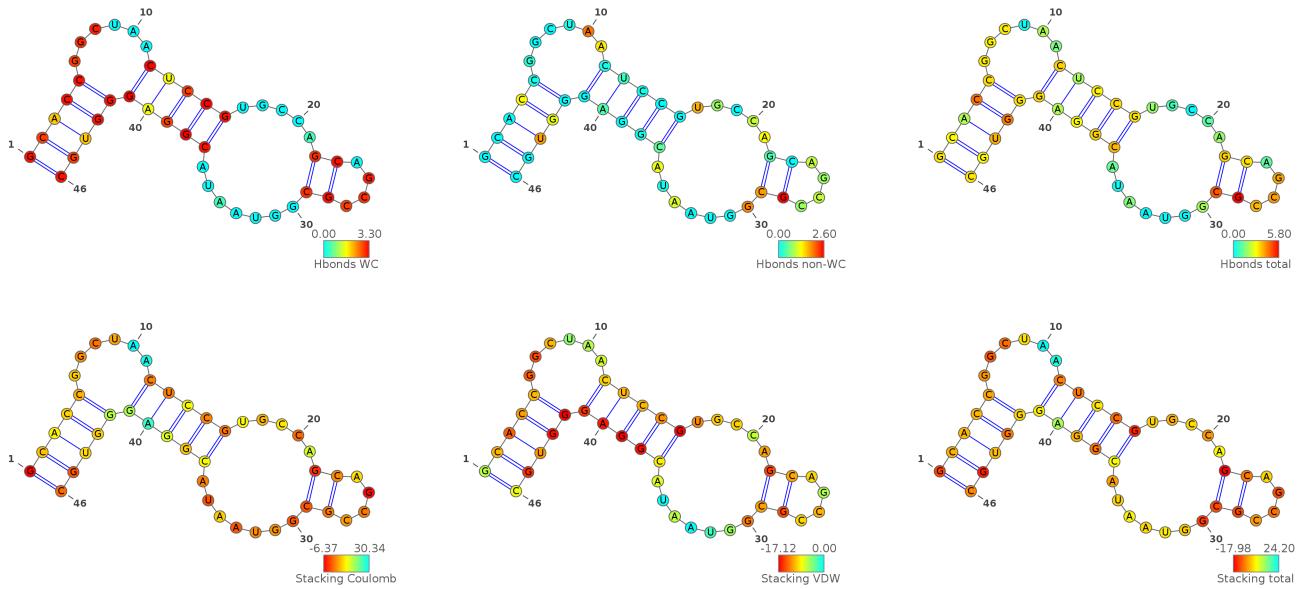


Figure 1: Figures with average secondary structure colored by the values of computed parameters, generated by `MINT` and displayed with the `VARNA` [7] program, respectively: number of WC hydrogen bonds, number of non-WC hydrogen bonds, total number of hydrogen bonds, Coulomb therm of Stacking energy in kcal/mol, Van der Waals term of Stacking energy in kcal/mol, sum of the Coulomb and Van der Waals interactions in kcal/mol. In case of energy (Coulomb and VDW) the color scale is reversed so the red nucleotides are the ones that are influenced by the strongest hydrogen bonding and stacking interactions.

5.6 Inputs for external visualization tools

Files needed for visualization using external tools (see Section 8 for details):

- _RNAStructML.xml
- _Hbonds-WC.pdb
- _Hbonds-non-WC.pdb
- _Hbonds-total.pdb
- _Stacking-Coulomb.pdb
- _Stacking-VDW.pdb
- _stacking-total.pdb
- _vmd_run.tcl

6 Methods

6.1 Workflow

Figure 2 shows the structure of the program. First, the program reads in the conformation and finds all hydrogen bonds between nucleotides. Using that, it determines secondary, tertiary structures and subsequently classifies the secondary structural motifs. The main outcome is a set of statistical descriptors of the given RNA structure.

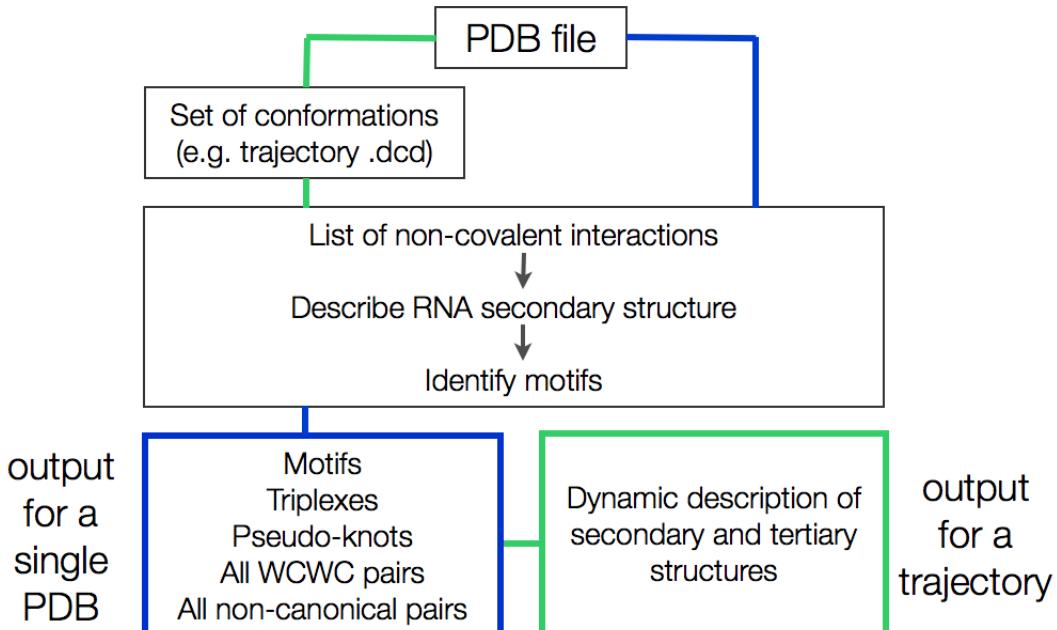


Figure 2: Main function implements the analysis of a single frame. In the case of a trajectory, the function reads-in the list of nucleotides from a given PDB and then refreshes the atom coordinates while reading sequential frames. Additionally, the script splits the trajectory between CPUs and runs separate processes, what accelerates the calculations.

6.2 Loading the DNA or RNA conformations

All of the scripts are written in Python programming language. Its modular structure enables applying it in various programming contexts. Reading the PDB file is implemented in BioPython package, providing a complete objective structure for dealing with PDBs.

In BioPython an atom is a basic object with fields for a name, number and spatial coordinates. Atoms build residues and residues chains, chains molecules. This allows fast and easy access to the chains, nucleotides, atoms and finally their coordinates.

6.3 Calculating physico-chemical features

6.3.1 Hydrogen bonding

Hydrogen bonds are basic interactions responsible for creating the secondary and tertiary structures of nucleic acids. A typical definition of a hydrogen bond pertains to a non-covalent interaction when a hydrogen atom, covalently bound to its donor, is placed close to its acceptor.

Criteria for detecting hydrogen bonding is based on the angle, created by donor, hydrogen and acceptor, and distance between acceptor and hydrogen or between donor and acceptor (Figure 3). Theory states that all hydrogen bonds are almost linear (around 175°) [10]. For biological molecules the hydrogen bond distance should be typically between 2,80 and 3,06 Å between a donor and acceptor, which gives 1,60 and 1,80 Å between an acceptor and hydrogen. The user can decide which distance will be measured by MINT using the `h_bond_atom` parameter.

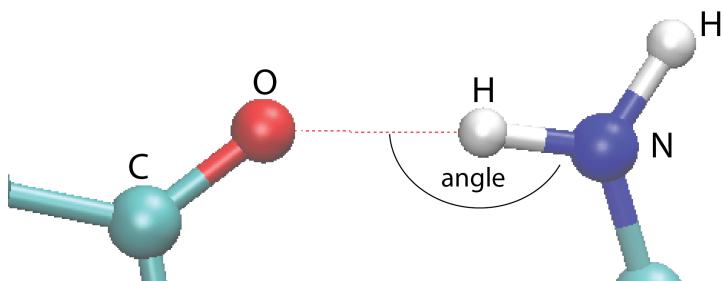


Figure 3: The scheme of the hydrogen bond with the nitrogen atom as a donor and the oxygen atom as an acceptor.

Other user defined parameters are: the minimal angle and the maximal length of a hydrogen bond. The default values of the `h_bond_distance` (3.5 Å) and the `h_bond_angle` (150°) are consistent with the distance calculated between donor and acceptor atoms.

It is possible that in the crystal structures or user-provided structures more than three hydrogen bonds in a GC pair (see Figure 4) or two in AU pair will satisfy the above criteria. The user can remove the "unwanted" hydrogen bonds by decreasing the distance and/or increasing the angle. However, one should have in mind that too strict criteria can cause errors in the analysis of nucleotide pairs and assignment of the RNA secondary structure.

6.3.2 Donors and acceptors

To analyze the structures we have defined a list of possible acceptors and donors for all nucleotides of RNA and DNA. Following the classification by Leontis and Westhof [13] the acceptors and

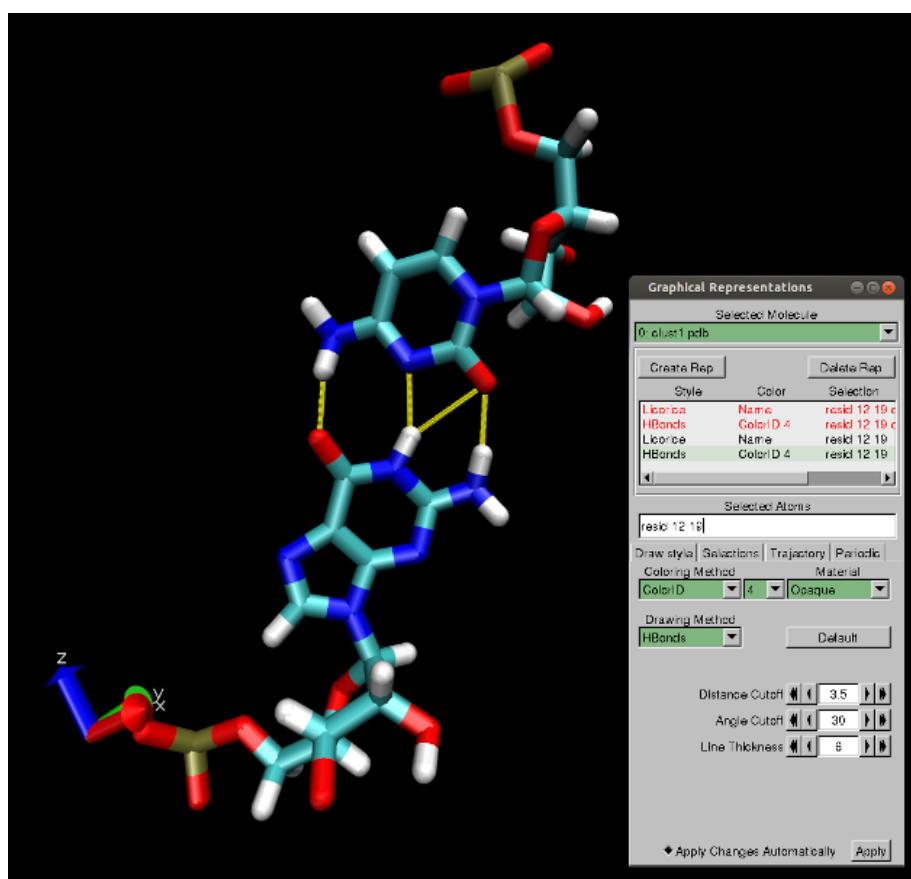


Figure 4: Visualization of four possible hydrogen bonds (yellow) found in the GC pair using default parameters for hydrogen bonds.

donors are assigned to the edges of the nucleotide (Figure 5). This classification is stored in the `table_nucleotides.csv` file and may be edited by the user.

MINT determines the interacting edges of every pair of nucleotides. Several atoms are situated in the corners of the nucleotides and participate in more than one edge. In that case, the program first classifies all the remaining bonds and chooses the prevailing edge. If there is only one hydrogen bond, both edge names are returned.

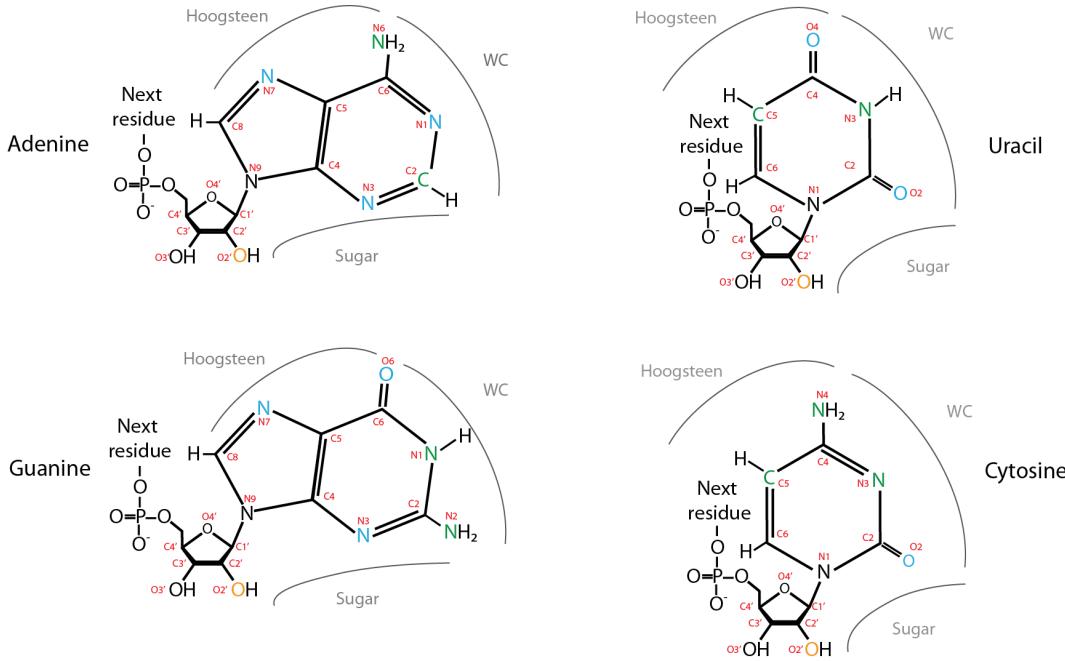


Figure 5: Nucleotides with selected edges, donors (green) and acceptors (blue). WC corresponds to WC edge. [14].

MINT checks all the donors against all acceptors in all possible pairs of all nucleotides in the molecule. In order to decrease the computational time, we assume that the partner for a given nucleotide can be found only among its closest neighbors. The exact distance is defined by the user in the `cutoff` parameter. Knowing the atoms creating hydrogen bonds, the program determines the interacting edges and classify the basepair.

Note that we use the edge-to-edge classification, instead of the concept of the canonical base pairs, to unambiguously and consistently describe all possible geometric pairs that may, even transiently, occur in a trajectory. The pair is classified as WC-WC if both nucleotides make hydrogen bonds with the WC edges. If there is only one bond between the nucleotides, the pair will be classified as WC-WC only if edges in both nucleotides can be unambiguously defined. For example if at least one of the interacting atoms is located on the "corner" between two edges (see: 5) the pair will be classified as non-WC-WC. If there are more than one hydrogen bonds between nucleotides than the edges are clearly defined and pair classification is straightforward.

6.3.3 Modified nucleotides

Modified nucleotides are found in almost all RNAs and are believed to be functionally important, for example, in tRNAs. A few modified nucleotides are presented in Figure 6.

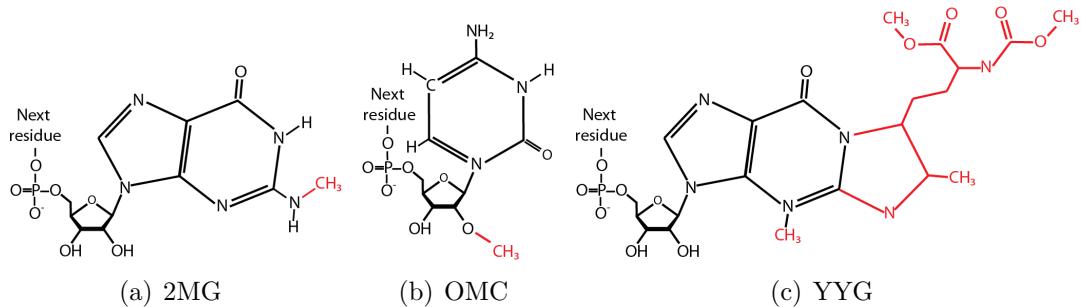


Figure 6: Three out of ten modified nucleotides present in the structure of tRNA (PDB code: 1TTT) along with their PDB names. The atoms that are not present in standard nucleotides are shown in red.

There are about 600 modified RNA nucleotides in the PDB database, but for most of them there are no force field parameters. With MINT we provide the VDW parameters and partial atomic charges for 107 naturally occurring modified nucleotides whose force fields were developed by Aduri et al [1]. For these nucleotides, we have also assigned their atoms to distinct edges: the WC edge, the Hoogsteen edge, the sugar edge and to the edge termed the modification edge (see Section 6.3.2 for details about the classification of edges).

An atom in a modified nucleotide, common to the unmodified one, is assigned into the same edge as in the unmodified nucleotide. The addition of no more than one heavy atom to the nitrogenous base is also classified into the same edge as the base atom. The 2’O methyl carbon is assigned to the sugar edge. All other atoms present in the modified nucleotide but not in the original one are assigned to the modification edge.

If MINT encounters a modified nucleotide, it looks for its parameters in the `table_nucleotides`. If parameters are not found, MINT uses the `list_of_modified_nucs` to find the natural ancestor of the modification. Atoms common to the modified and its natural ancestor are assigned analogously as in the original nucleotide, the rest is assigned to the modification edge. If there are no partial charges and VDW parameters, the stacking interaction energy is set to 0.

However, the user can add new modified nucleotides. The force field parameters developed for a modified nucleotide for an MD simulation, can be adopted to the MINT parameter format. If you want to add your own modification to the MINT parameters: you should edit the `table_nucleotides` file and add a new row with the name of your residue and appropriate names of atoms in the donor and acceptor columns. Next, you should also modify the `table_charges` file. For every atom that should be considered in stacking computation you have to add a row with the name of the residue, charge, VDW radius and well depth parameter. For details see Sections 6.3.1 and 6.3.5.

6.3.4 Geometric isomerism of base pairs

For detected base pairs the geometric isomerism of their glycosidic bonds is computed. The program measures the torsion angle formed by the four atoms (C1’, N1 in pyrimidines and C1’, N9 in purines) and depending on its value a *cis* or *trans* conformation is denoted (see Figure 7).

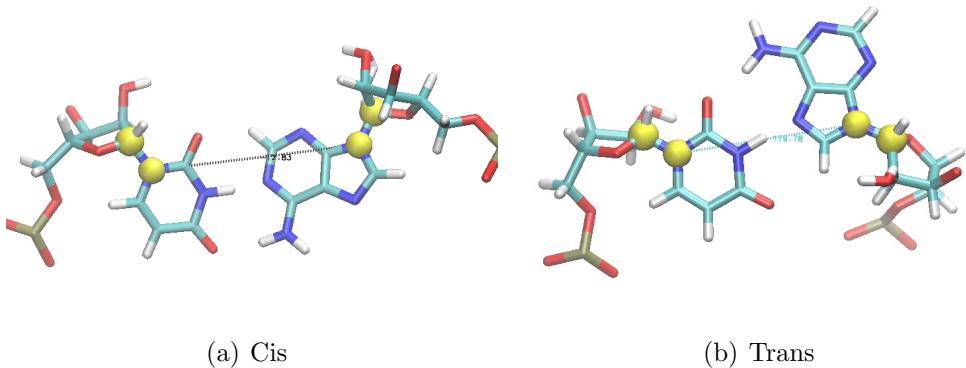


Figure 7: Yellow spheres correspond to C1' and N1 atoms in pyrimidines and N9 atoms in purines. The torsion angle created by these four points determines the geometric isomerism of the two nucleotides creating a pair.

6.3.5 Stacking

Stacking is an important non-covalent interaction contributing to the stability of both double helix and single stranded structures of nucleic acids [11]. For instance, in tRNA only half of the nucleotides form a helix but about 90% are stabilized by stacking [4]. Generally, stacking occurs

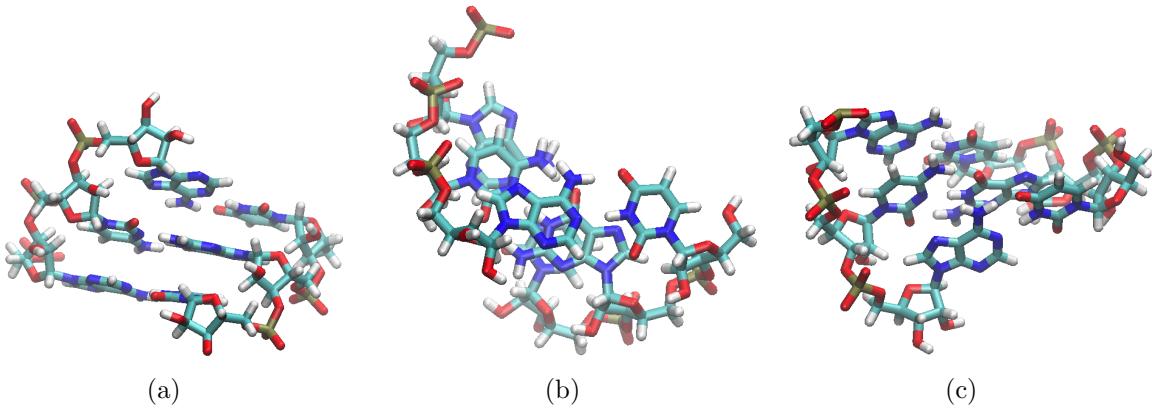


Figure 8: Three-pairs of an RNA helix seen from different angles to expose stacking between parallel bases.

between aromatic rings so in nucleic acids between the nucleobases. There is a general belief that stacking results from the contact of the electron π -systems. Stacking also arises from three phenomena: Van der Waals (dipole or induced-dipole attractions), electrostatic and solvation effects. Stacking interactions seem to be more important in folding of nucleic acids than proteins because nucleobases are more polarizable than most amino-acids.

Stacking is believed to be represented well in molecular modeling, especially with the Amber atomic charges which are fitted to molecular electrostatic potentials [11]. It was shown that calculations using the empirical potentials consisting of the Lennard-Jones VDW and Coulombic

terms with atom-centered point charges reproduced the *ab initio* stacking energy over the major portion of the conformational space [15]. Šponer et al. in many studies [3, 5, 22, 23] compared *ab initio* energies for about 300 geometries of stacked base dimers with the data obtained using empirical potentials. The agreement between these methods is remarkable, which suggests that calculations based on empirical potentials provide an excellent approximation of the stacking interaction energy between nucleotides. Therefore, MINT calculates the stacking interactions.

We estimate stacking between two bases by calculating the energy of electrostatic (U_{el}) and Van der Waals (U_{VDW}) interactions applying the equations used in molecular mechanics:

$$U_{el} = k \sum \frac{q_i q_j}{r_{ij}} \quad (1)$$

$$U_{VDW} = 4\epsilon_{ij} \sum \left[\frac{1}{4} \left(\frac{r_0}{r_{ij}} \right)^{12} - \frac{1}{2} \left(\frac{r_0}{r_{ij}} \right)^6 \right] \quad (2)$$

The sums run over all atom pairs of nucleobases i and j , k denotes the Coulomb constant ($k = \frac{1}{4\pi\epsilon_0}$), q is the partial atomic charge of the atom, r_{ij} is the distance between the considered atoms, ϵ_{ij} is the depth of the Lennard-Jones potential well for atoms i, j ($\epsilon_{ij} = \sqrt{\epsilon_{ii}\epsilon_{jj}}$) and r_0 is the sum of VDW radii of atoms i and j . We provide the VDW well depth parameters and partial atomic charges from the Amber [24] and Charmm [9, 16] force fields. A set of parameters and charges may be also defined by the user in the file `table_charges` file. Only the nucleobases that are closer than the user defined cutoff (`cutoff_stacking`) are considered in the stacking calculations. The energy unit obtained from described calculations is *kcal/mol*.

The electrostatic term, depending on the orientation of base dipole moments, may be attractive or repulsive regardless of the mutual orientation of the bases (parallel or not). While the VDW energy component is almost always favorable regardless of the orientation of bases. Moreover, the shape of nucleotides and the method used for calculating the VDW energy ensure that the lowest VDW energy values are obtained for the parallel orientation of nucleobases (with their largest overlap). This was also the case in our test calculations. Thus we recognize two nucleotides as stacked, if their VDW energy is lower than the `vdw_cutoff_stacking` parameter. The default value of this parameter is -0.5kcal/mol which was found by trial and error and seems appropriate for the nonmodified nucleobases.

6.3.6 Ion-II interaction between a nucleobase and phosphate oxygen

Hydrogen bonds are the most known non-covalent interactions but are not the only one. For example, the cation-II interactions, namely the non-covalent bonding between a monopole (cation) and a quadrupole (II system), play a role in the structure of proteins. A similar interaction was reported for RNA but it was found that nucleic acid aromatic systems prefer to interact with anionic rather than cationic species [2]. MINT enables searching for anion-II interactions involving the RNA backbone phosphate groups and nucleobases. An example of such a contact is shown in Figure 9. The interacting systems are recognized if the distance between a phosphate atom and nucleotide base center of mass is lower than the `OP_stacking_distance_cutoff` parameter (default distance is 5 Å). If the ion-II stacking is detected, the energy of interaction between the phosphate atom and nucleobase is calculated in the same way as for the stacking between two nucleobases.

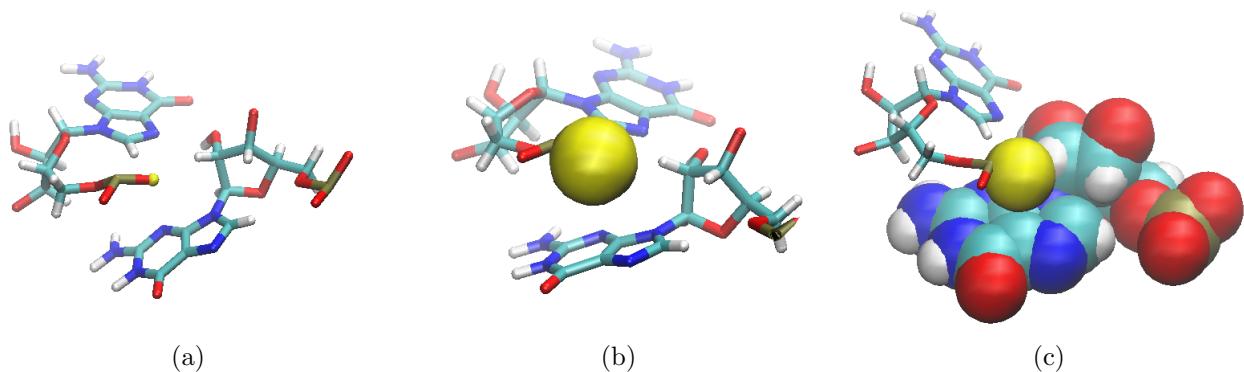


Figure 9: Three representations of the oxygen atom (yellow) ”stacking” over the guanine base. The yellow sphere in the picture corresponds to the real VDW radius.

6.3.7 Representation of RNA motifs

Having all WC pairs, a list-representation of the RNA secondary structure is created. Most nucleotides have only one WC-edge partner but in MD trajectory it may transiently happen that a second WC-edge partner is encountered, and such a triple is not considered in the secondary structure analysis. The index of the list represents the nucleotide number and the stored value is the index of its WC partner. The list is easy-interpretable when the arcs connecting the pairs are drawn as presented in Figure 10.

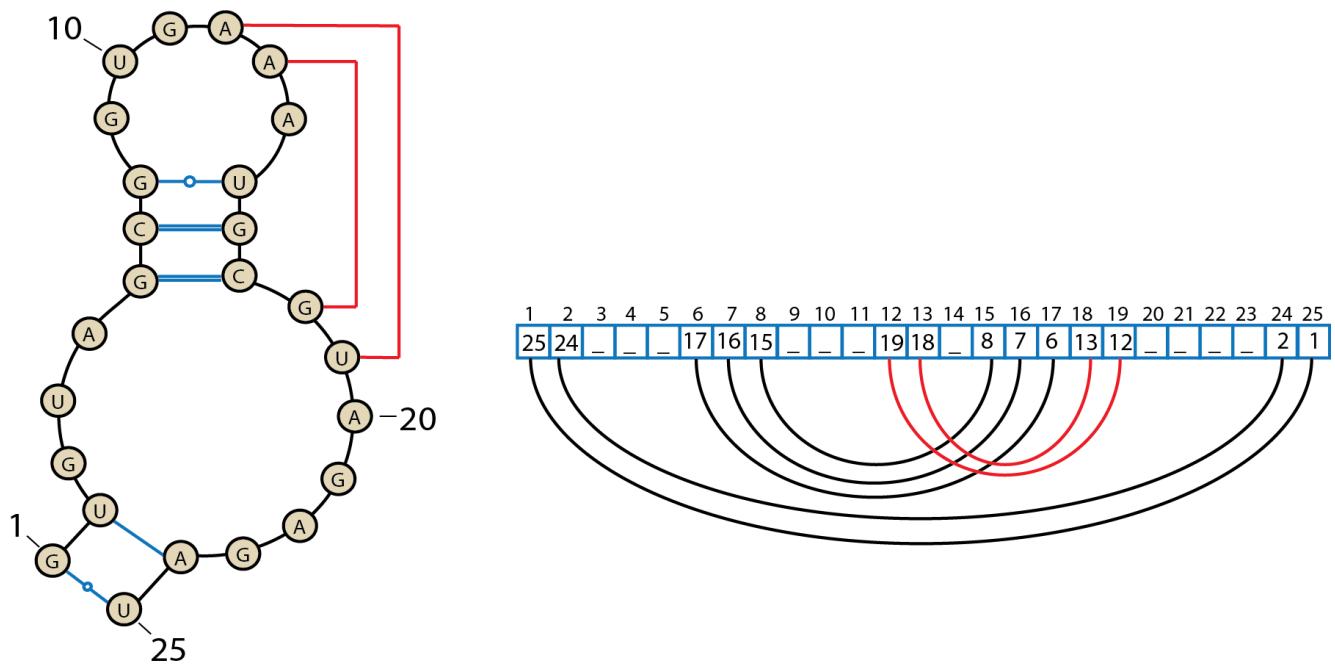


Figure 10: Secondary structure of an exemplary RNA molecule and its list-representation. Red lines correspond to the WC interactions creating a pseudoknot.

Pseudo knots The list-representation contains also the information about the non-secondary motifs. The pseudoknot is formed by the WC interactions but creates three-dimensional folds as shown in Figure 11. Our program detects the pseudoknot fold when the arcs intersect. A pseudoknot is a symmetric structure so both the pairs 6–17, 7–16, 8–15 and 12–19, 13–18 in Figure 10 form a pseudoknot. The natural way of solving this conflict is to choose the shorter list, in this case the pairs 12–19 and 13–18. To erase pseudo-knots from the list-representation of the secondary structure, so they do not disturb the motif-search algorithm, we use a conflict elimination method [21] leading to a nested structure containing the maximum number of base pairs.

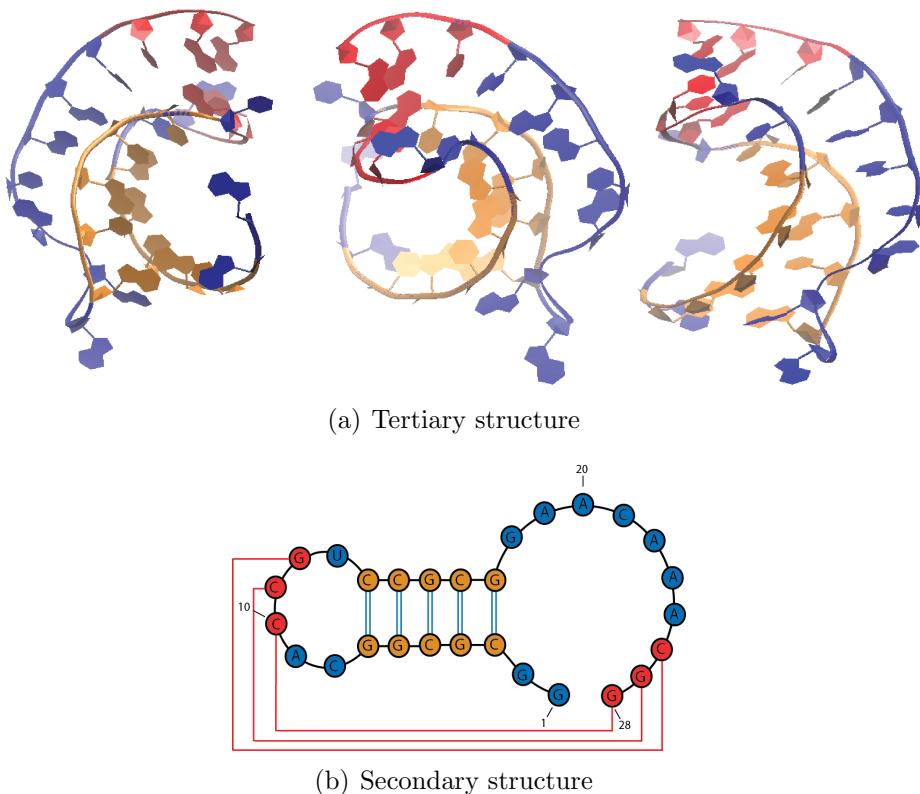


Figure 11: An example of the RNA structure with a pseudoknot seen from three different angles. Nucleotides colored in red create a pseudoknot, orange form a helix and blue refer to loops (PDB code: 437d).

After detecting all pairs and creating a list, MINT finds all pseudoknots and erases them from the list by putting the `None` value. Next, it looks for all other kinds of motifs, that are defined as a set of unpaired nucleotides surrounding the paired nucleotides as shown in Figure 12.

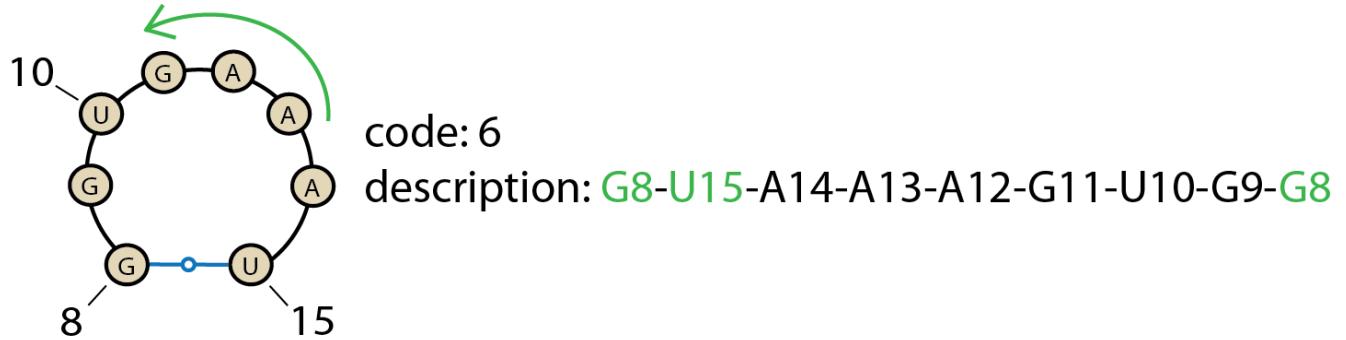
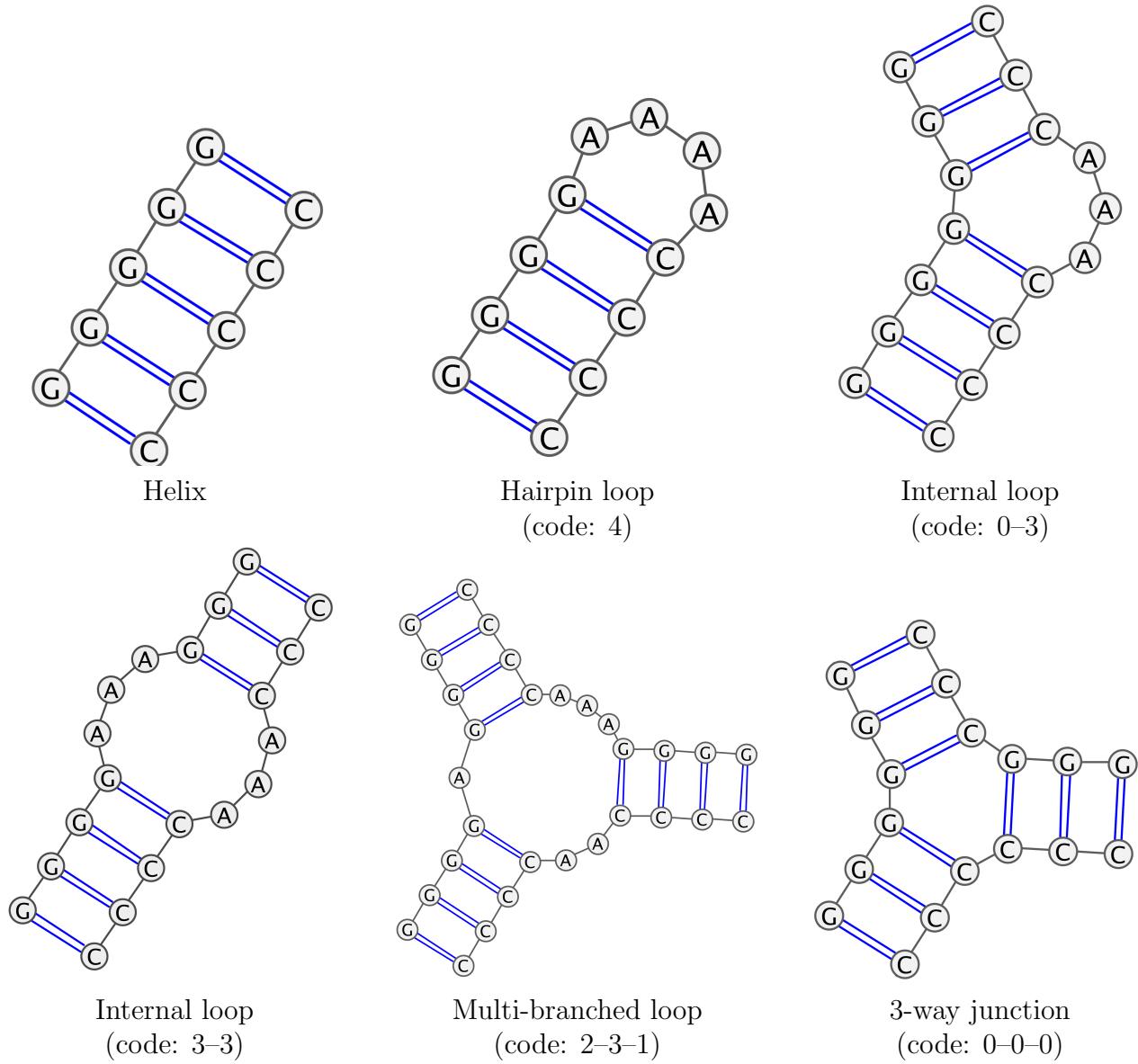


Figure 12: The program denotes the first pair (shown in green) and then makes a list of unpaired nucleotides. Nucleotides are listed in the counter-clockwise direction.

Table 8: RNA secondary structure motifs



The RNA motif nomenclature (i.e., a loop, bulge or junction) can be inaccurate and misleading. The program uses the numerical description of motifs that should be easily understandable. The numbers correspond to the numbers of unpaired nucleotides between the paired nucleotides. Therefore, a four nucleotide loop at the end of a hairpin is represented by a single number 4. A symmetric bulge loop, with both bulges of the length of three is represented by two numbers: 3–3 and a three-way junction, without unpaired nucleotides is represented by three zeros: 0–0–0. Examples of RNA secondary structure motifs with their labels are presented in Table 8.

6.3.8 Motif-search algorithm

The motif-search algorithm uses a list-representation of the RNA-secondary structure that contains the list of all WC pairs. As shown in Figure 13 the algorithm detecting helices and other secondary structure motifs walks through the list and stores the information about the visited nucleotides.

As long as there are paired nucleotides, the algorithm stores the information about the helix (steps 1 and 15 in Figure 13). When it encounters the end of the helix (steps 2 and 16) – i.e., an unpaired nucleotide ahead of a pair, it starts to travel (search) around the motif. It stores the first pair (as the beginning pair) and goes to the index stored in the list. Then for as long as it encounters an unpaired nucleotide it moves back – with decreasing indexes (steps 3 to 13 and steps 16 to 22). When another pair is encountered, the algorithm goes into the indicated position and again moves back (steps 10 and 16). This motif-search ends when the algorithm finds itself one step ahead of the starting index (steps 13, 22). The number of "jumps" represents the number of unpaired nucleotides in between. After one motif is found and classified, the algorithm jumps one step further to the last seen pair (steps 14 and 23). The algorithm stops searching for the motifs and helices when the index is larger than the value stored in the list (step 23).

As a result of a single frame analysis, the algorithm returns a list of all pairs, motifs and numerical characterization of all nucleotides. Every nucleotide is characterized with the number of hydrogen bonds it created and the description of its partners:

G538 , 538 , 3-C513A539:1 , 2-C513:2 , 3-C513:7

where first comes the type of the nucleotide with its PDB code, second only the PDB code, followed by the lists of pairs this nucleotide created during the trajectory or in the uploaded conformation set. The number of hydrogen bonds is preceding the hyphen. Next, there can be more than one nucleotide listed – what indicates a triplex. The number after a semicolon is the number of frames in which this pair or triplex were detected.

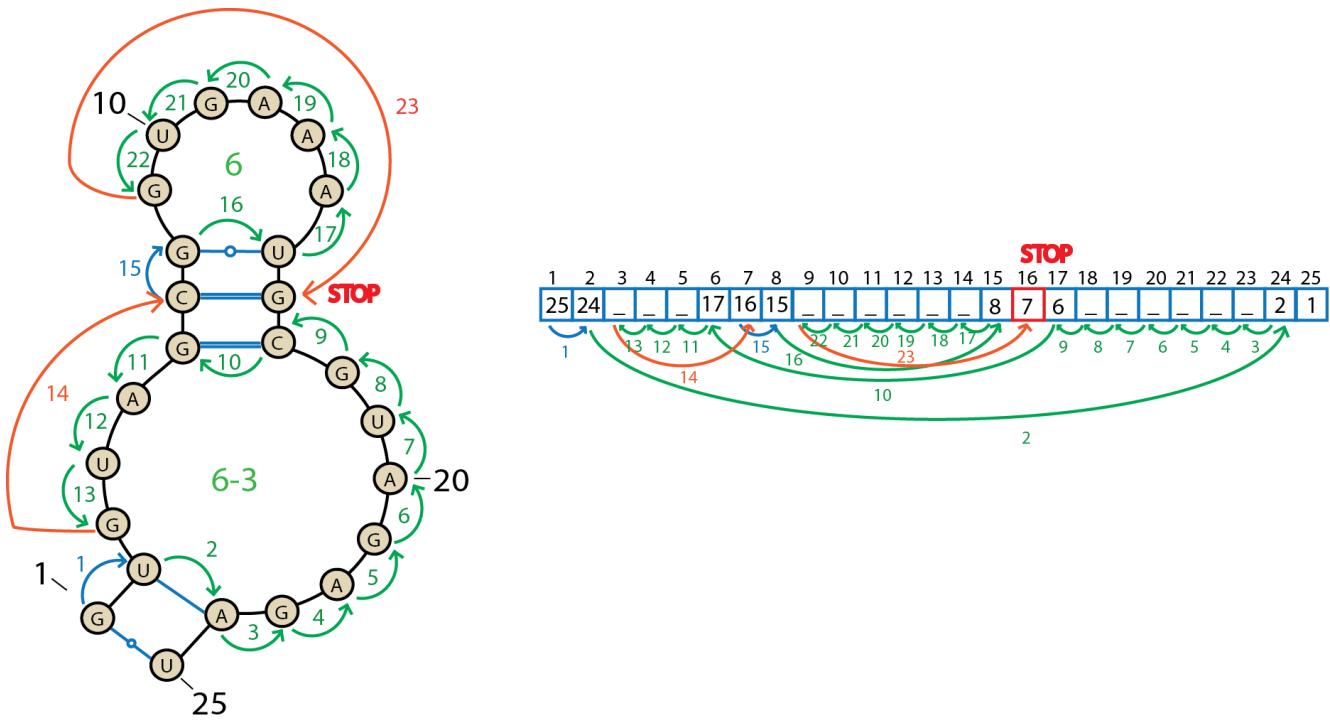


Figure 13: A scheme presenting the steps of the algorithm that finds and classifies the motifs in the RNA structures. The numbers correspond to the step number. Green arrows correspond to the structural motif and blue to helices. The orange lines denote the jumps after finishing the motif. The STOP sign indicates the position where the algorithm stops.

7 Analysis of trajectories or many RNA/DNA conformations

If multiple RNA/DNA conformations (e.g. from molecular dynamics trajectory) have to be analyzed and classified, every frame/conformation is characterized as previously described. The main output of the program is a set of .csv files listing all the pairs and motifs: helices, triplexes, pseudoknots etc., their occurrence and frame numbers in which the considered structure was present, its topology and participating nucleotides. Description of the output files can be found in Section 5.

7.1 Clustering

To recognize and characterize the changes in the secondary structure of the RNA/DNA molecule that may occur in the trajectory, we have to cluster the detected secondary structure motifs. Clustering is parameterized with two user-defined parameters:

- `time_cutoff` that defines the minimal percentage of frames in which the motif has to be present in order to be classified.

- **margin** that defines the minimal percentage of similarities between the two motifs to belong to the common cluster.

The first step is to filter the motifs i.e., remove the rare motifs and leave only the significant ones. Filtering of motifs is done according to the number of frames the motif appeared in. Below we show an exemplary list of motifs after removing rare motifs:

1. C13-G29-U28-G27-C15-A14-C13-
2. C2-G42-C41-G40-C4-G3-C2-
3. C8-G35-C34-U33-G32-C11-U10-G9-C8-
4. C1-G43-G42-C41-G3-C2-C1-
5. C11-G32-A31-A30-G29-U28-A14-C13-A12-C11-
6. G9-C34-U33-G32-C11-U10-G9-
7. A14-U28-G27-G26-C16-C15-A14-
8. C11-G32-A31-A30-G29-C13-A12-C11-
9. G9-C34-U33-G32-A31-A30-G29-C13-A12-C11-U10-G9-

Figure 14: First step of clustering.

Second, the motifs' distance matrix is computed. The distance between the motifs is the number of their common nucleotides. The order of the nucleotides is not taken into account:

motifs	1	2	3	4	5	6	7	8	9
1	6	0	0	0	4	0	4	2	2
2	0	6	0	4	0	0	0	0	0
3	0	0	8	0	2	6	0	2	6
4	0	4	0	6	0	0	0	0	0
5	4	0	2	0	9	2	2	7	7
6	0	0	6	0	2	6	0	2	6
7	4	0	0	0	2	0	6	0	0
8	2	0	2	0	7	2	0	7	7
9	2	0	6	0	7	6	0	7	11

Figure 15: Exemplary motif distance matrix

Third, the partners for all motifs are denoted. The partner has to satisfy the distance requirement expressed by the equation shown below:

```
if T[i][j]>threshold*min(len(i), len(j))
threshold parameter is defined by the user.
For threshold = 0.6 the first list of friends
looks like this:
```

1. 1 5 7	5. 1 5 8 9
2. 2 4	6. 6 3 9
3. 3 6 9	7. 7 1
4. 4 2	8. 8 5 9
	9. 3 5 6 8 9

Figure 16: Equation with the distance requirement that motifs have to satisfy to be partners

Next, the motif with the longest list of partners is incorporated as the first one to the first cluster (numbered zero). The motifs already assigned to the first cluster have to be crossed out from the rest. Then the second longest motif is chosen, the picked motif is crossed out, and so on.

1. 1 5 7	5. 1 5 8
2. 2 4	6. 6 3 9
3. 3 6 9	7. 7 1
4. 4 2	8. 8 5 9
	9. 3 5 6 8 9

repeat step 3 as long as there are
motifs in the list ..

Figure 17: Removal of partners from the list of motifs

Additionally, the average clusters are returned. The list consists of the longest motif for every cluster and the two vectors of float values describing every nucleotide participating in the representative motif. The float vectors represent the average number of hydrogen bonds created by a nucleotide in both WC-WC pairs and other interactions. We hope to show the average secondary structure of the molecule within the tertiary structure description. While analyzing the whole output generated by the script, one can understand the complex description of the RNA structure in the dynamics.

Output cluster list:

cluster 0	cluster 2
8. C11-G32-A31-A30-G29-C13-A12-C11-	1. C13-G29-U28-G27-C15-A14-C13-
5. C11-G32-A31-A30-G29-U28-A14-C13-A12-C11-	7. A14-U28-G27-G26-C16-C15-A14-
6. G9-C34-U33-G32-C11-U10-G9-	
3. C8-G35-C34-U33-G32-C11-U10-G9-C8-	
9. G9-C34-U33-G32-A31-A30-G29-C13-A12-C11-U10-G9-	
cluster 1	
2. C2-G42-C41-G40-C4-G3-C2-	
4. C1-G43-G42-C41-G3-C2-C1-	

Figure 18: Result of the clusterization process.

8 Visualization

MINT enables many different ways of data visualization. The user can display a colored input PDB structure according to several parameters (motifs, secondary and tertiary contacts, electrostatic energies, VDW energies) both in the **Single** and **Trajectory** mode.

For visualization we suggest to install the following programs:

- VMD [12], or **Chimera** [18] for tertiary structure visualization.
- **RNAMovies** for secondary structure visualization changes in the trajectory (<http://bibiserv.techfak.uni-bielefeld.de/rnamovies/>).

8.1 Representing motifs on the tertiary structure

This feature (available only in the **Mode: Single**) works through creating representations of the regions of the nucleic acid molecule in the input PDB file using the VMD program [12]. In VMD one can represent fragments of a molecule using different representations and colors, all can be managed from **Graphical Representations** menu (**Graphics > Representations**). The `vmd_run.tcl` script loads the structure and creates **Reps** for every motif and helix in the structure what results with a view of the three-dimensional molecule colored accordingly to the detected structural components:

- helices: yellow (VMD color code:4)
- pseudoknots: tan (VMD color code:5)
- triplexes: silver (VMD color code:6)

- loops: green (VMD color code:7)

If the VMD program is in your PATH you can turn on the `vmd` parameter and it will launch automatically. Otherwise, you can do it manually by going into your MINT inputs/outputs directory and typing in the terminal:

```
your_vmd_location\ vmd -e _vmd_run.tcl
```

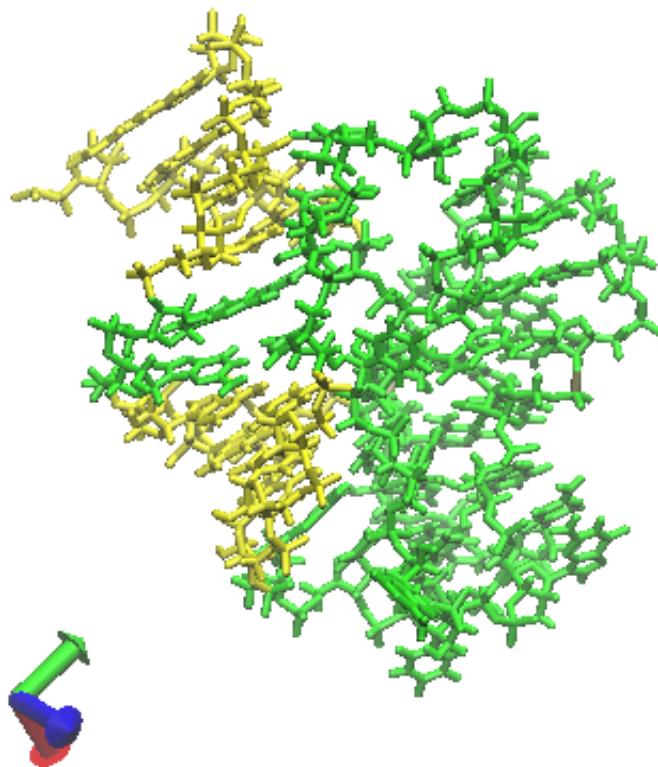


Figure 19: The result of running the exemplary structure `_vmd_run.tcl` script in the VMD program after changing the display into `Orthographic` and the background color to white.

8.2 Representing hydrogen bonding and stacking on the tertiary structure

The MINT program outputs several PDB files each with the B-factor column replaced for each nucleotide atom line with:

- `_Hbonds-WC.pdb` – the number of hydrogen bonds in the WC pairs created by a given nucleotide,
- `_Hbonds-non-WC.pdb` – the number of hydrogen bonds in the non-WC pairs created by a given nucleotide,
- `_Hbonds-total.pdb` – the number of hydrogen bonds in both WC and non-WC pairs created by a given nucleotide,

- `_Stacking-Coulomb.pdb` – the Coulomb energy for a given nucleotide (the sum of all electrostatic interactions the nucleotide is involved in),
- `_Stacking-VDW.pdb` – the Van der Waals energy for a given nucleotide (the sum of all interactions the nucleotide is involved in),
- `_Stacking-total.pdb` – the sum of the Van der Waals and Coulomb energies.

For a trajectory these PDB files contain the average values (over all frames). Here, we describe how to visualize the data outputted by MINT using the VMD or Chimera.

VMD VMD is a powerful tool with a complete user guide and tutorial that can be found at (<http://www.ks.uiuc.edu/Research/vmd/>). First, the user has to open the VMD program and load one of the PDB files, by choosing the **New Molecule** from the **File** drop-down menu as shown below:

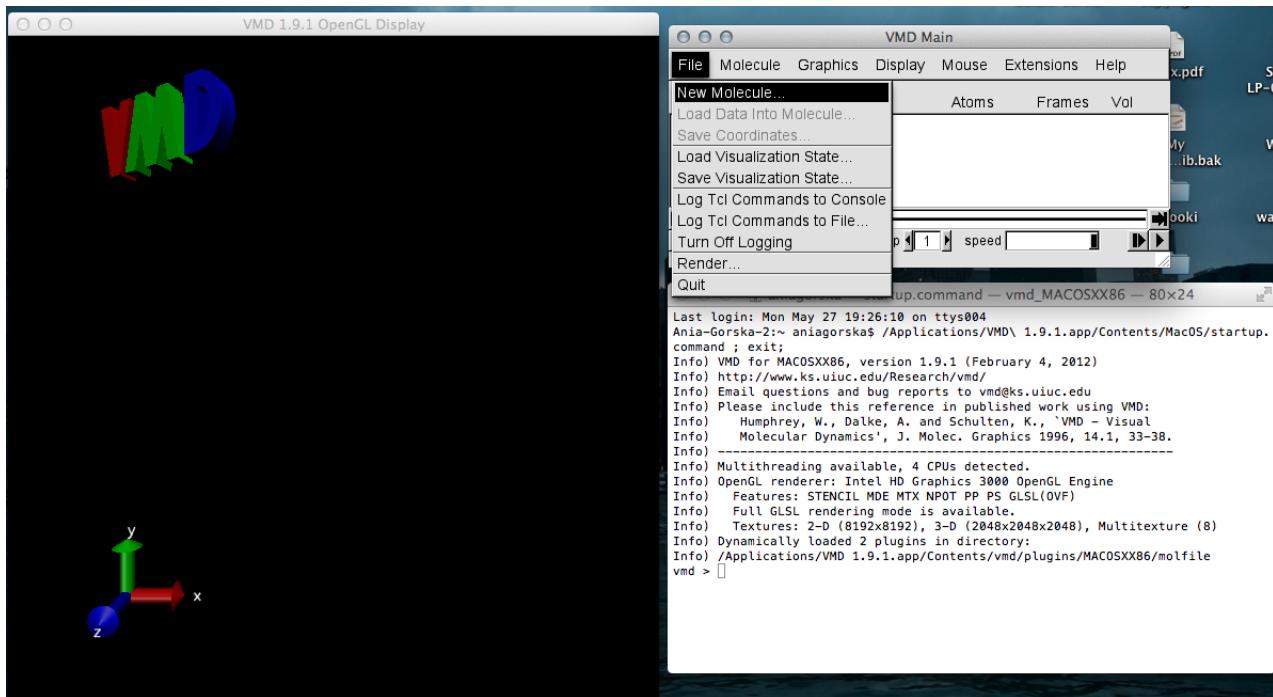


Figure 20: Loading a PDB file into the VMD program

Second, one has to **Browse** and **Load** a PDB file of choice:

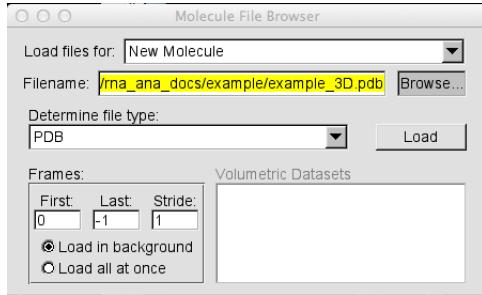


Figure 21: Loading a PDB file into VMD program

Having uploaded the structure, one needs to open the **Representations** window from the **Graphics** drop-down menu:

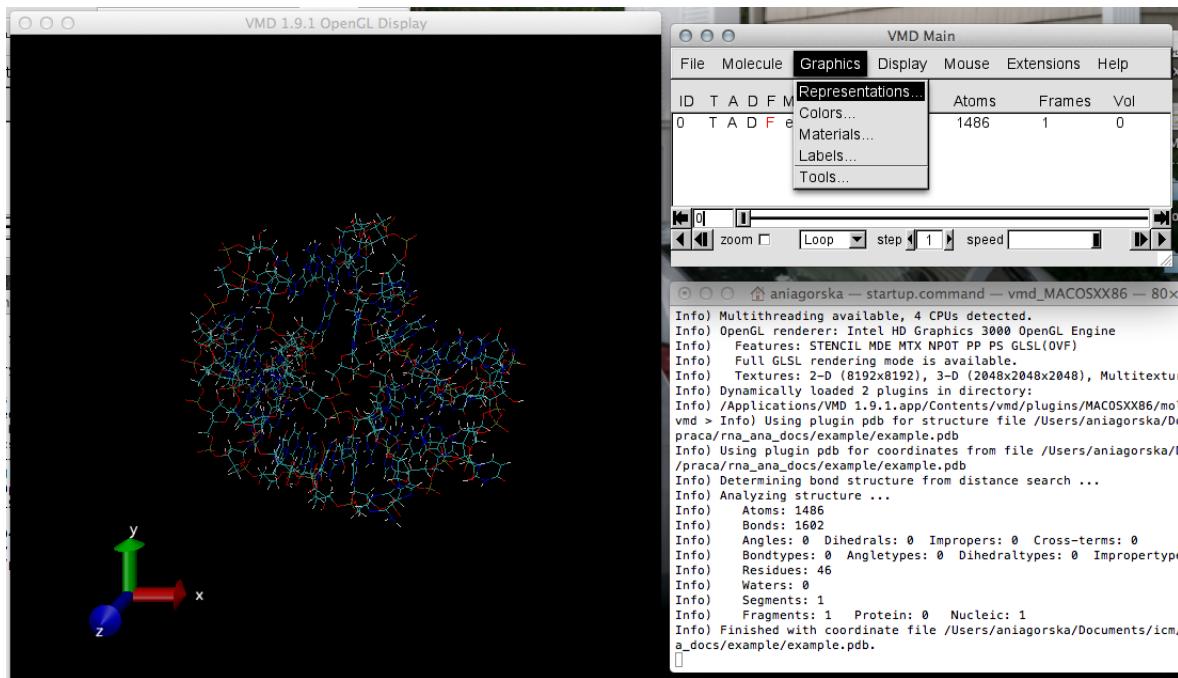


Figure 22: Opening the representation window in VMD

The **Representations** menu allows one to create various representations. To color the structure by the B-factor column in the PDB file, we suggest to change the **Drawing Methods** into the **New Cartoon** and the **Coloring Method** into the **Beta**:

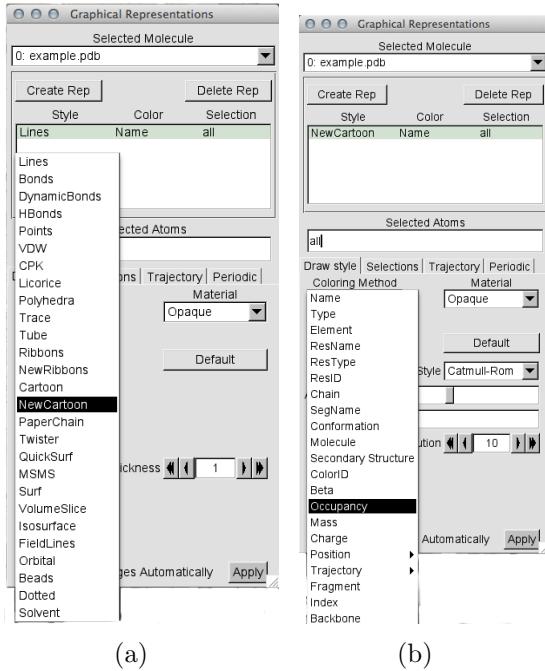


Figure 23: Choosing Drawing and Color Methods in the VMD program

Next, the color scale can be altered through **Graphics > Color > Color Scale**:

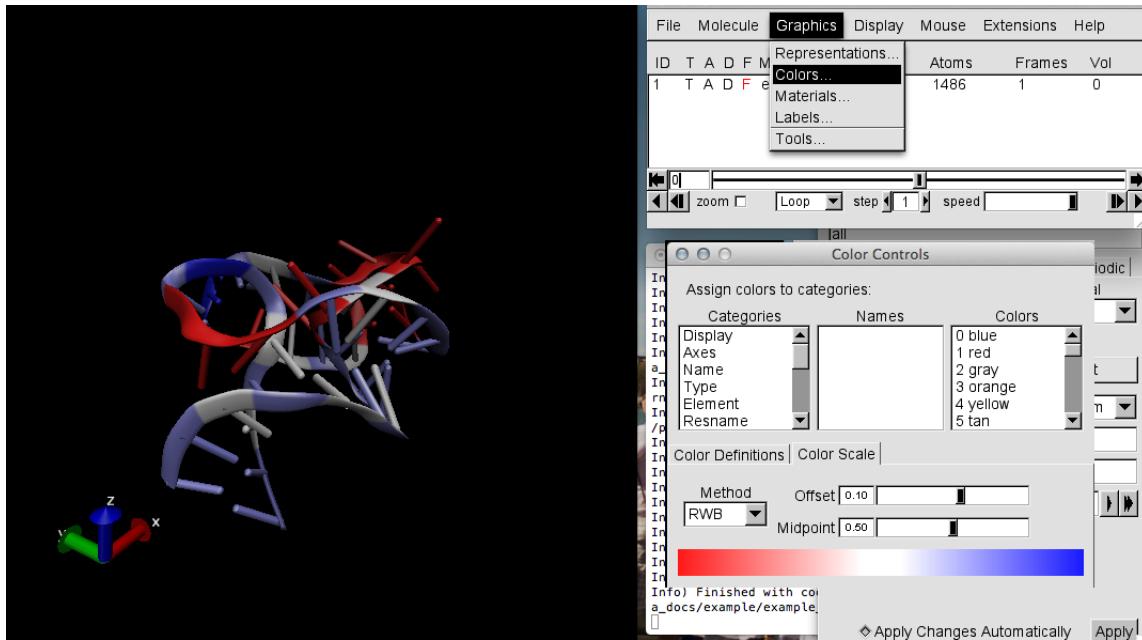


Figure 24: Changing the color scale

If you proceed in the same way with more than one PDB structure you can use the **Move** tool (**Mouse > Move > Molecule**) and view the colored structures at the same time as it is shown in Figure 25.

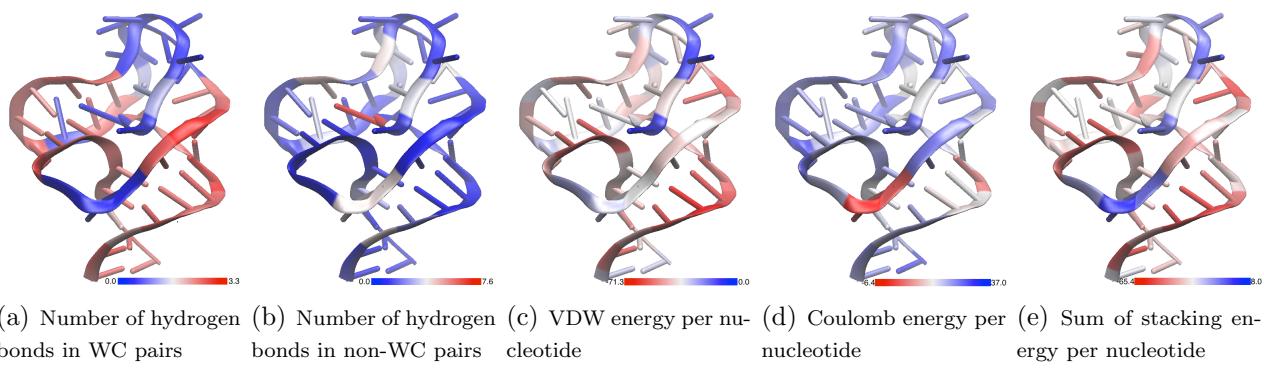


Figure 25: The starting tertiary structure rRNA fragment (nucleotides 500 to 545) colored according to various computed descriptors averaged from the trajectory (the same exemplary fragment was shown before in Figure 1). Figures created with the use of the VMD program and the MINT output PDB files. All energy is expressed in kcal/mol.

Chimera High-quality 3D visualizations can be also prepared with the Chimera software (details on how to use the program can be found at (<https://www.cgl.ucsf.edu/chimera/>)). The user has to open Chimera and load one of the PDB files and next, choose from the Tools menu the Depiction section and click on the Render by Attribute as shown in Figure 26.

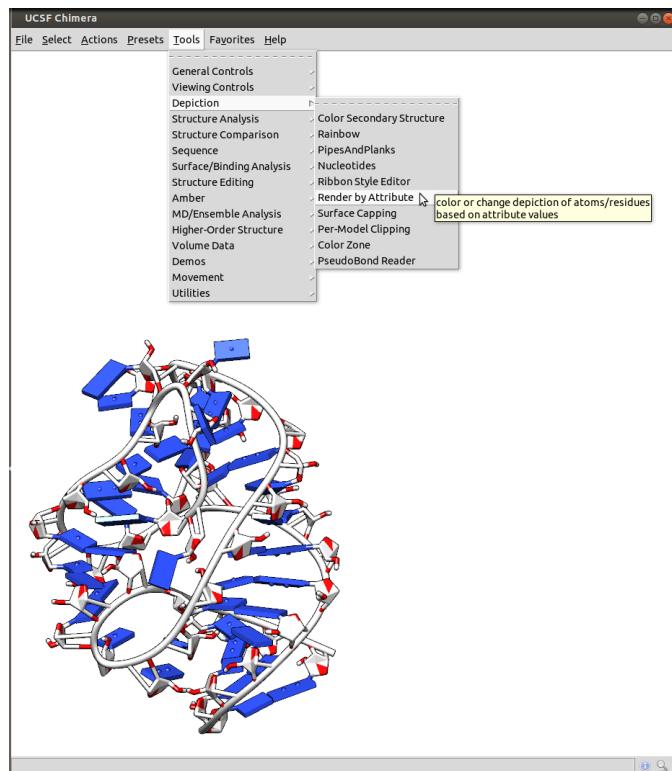


Figure 26: Render by attribute menu in Chimera

In the opened window (Figure 27) the user chooses attributes of `residues` instead of default `atoms`. The `Attribute:` should be changed to `average -> bfactor`. Then the color scale and palette can be chosen and the color key can be created. Two exemplary pictures rendered in Chimera are presented in Figure 28.

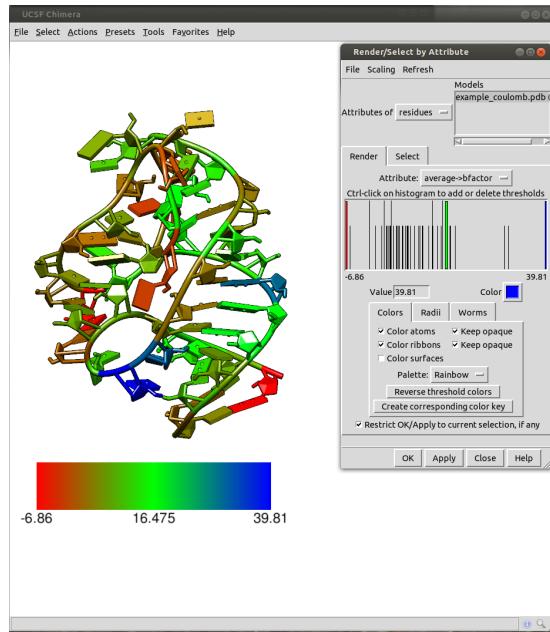


Figure 27: Rendering the structure by attribute in the Chimera program using the PDB file created by MINT

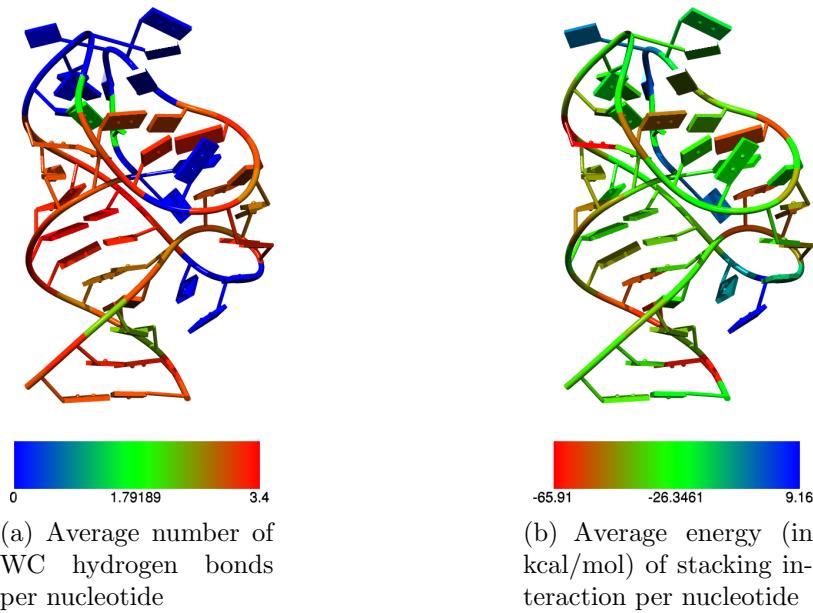


Figure 28: Exemplary pictures created with the use of the Chimera program and the PDB files created by MINT

8.3 Representing secondary structure changes

Analysed trajectory can be visualized in the secondary structure representation with the RNAMovies [8] program. MINT returns the `RNAStructML.xml` file containing the trajectory of the secondary structure. For every frame the dot-bracket representation is written into the .xml file. This allows to produce the movie of the secondary structure trajectory.

The RNAMovies [8] java .jar file can be downloaded from its home page: <http://bibiserv.techfak.uni-bielefeld.de/rnamovies/>. Here, we present a short tutorial on how to open an .xml file generated with MINT and view the secondary structure trajectory. First, one has to open the RNAMovies and choose the `Import` option from the `File` drop-down menu.

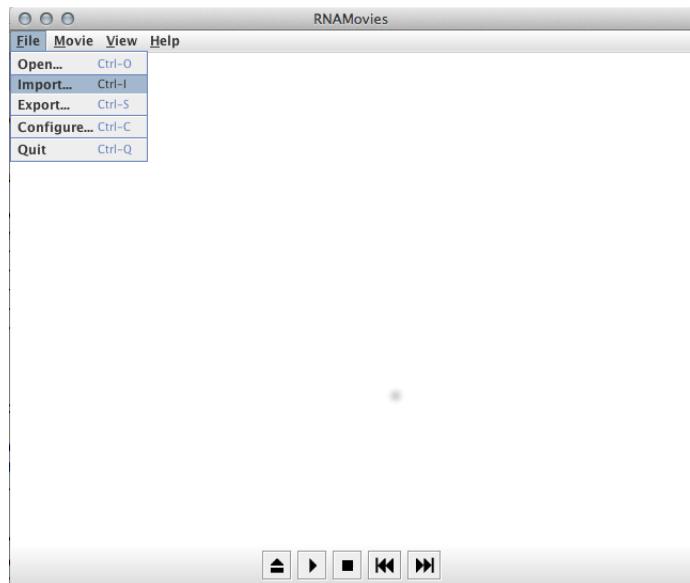


Figure 29: The RNAMovies window ready to load the .xml file

Find a proper .xml file in your directory, and click Open:

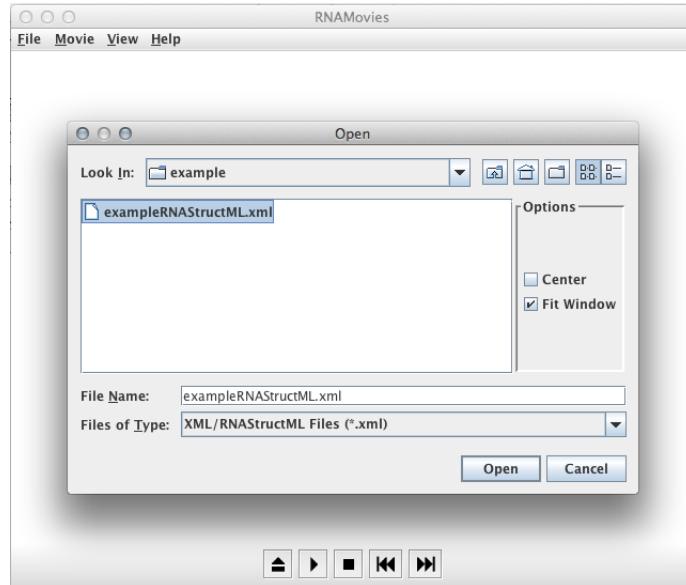


Figure 30: Loading the .xml file to the RNAMovies.

One can navigate the trajectory using the arrows in the bottom of the window. If you would like to loop the trajectory or change the animation speed go to the **File > Configure** menu:

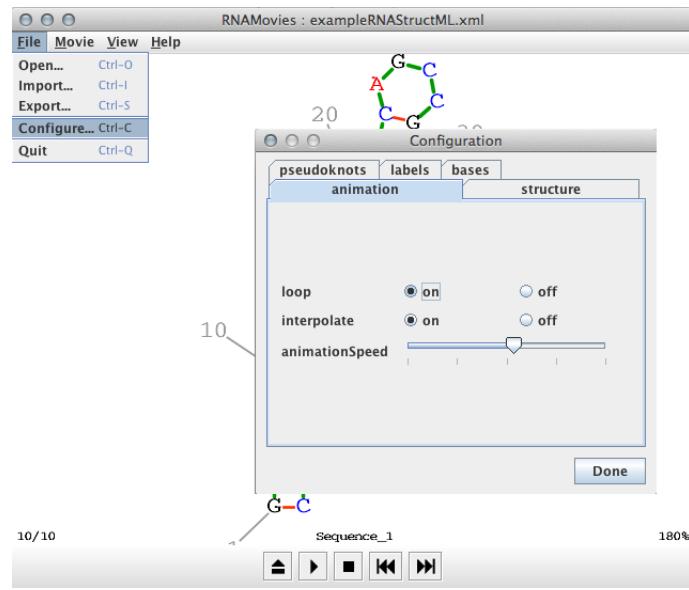


Figure 31: Adjusting properties of the animation in the RNAMovies.

The animated trajectory can be written into the .gif file (**File > Export**).

8.4 Correlations

In the MINT directory there is an additional python script `CORRELATIONS.py` computing the ϕ coefficient for every pair of nucleotides in the structure. This coefficient is computed using the equation [20]:

$$\phi = \frac{n_{11}n_{00}}{\sqrt{n_{\bullet 1}n_{\bullet 0}n_{0\bullet}n_{1\bullet}}}$$

where:

- n_{11} is the number of frames in which both nucleotides form a WC pair.
- n_{00} is the number of frames in which none of the nucleotides forms a WC pair.
- n_{01} is the number of frames in which the first nucleotide forms a WC pair and the second nucleotide does not, analogously n_{10} is the number of frames in which the first nucleotide is paired and the second one is unpaired.

and:

- $n_{\bullet 1} = n_{11} + n_{01}$
- $n_{1\bullet} = n_{11} + n_{10}$
- $n_{\bullet 0} = n_{00} + n_{10}$
- $n_{0\bullet} = n_{00} + n_{01}$

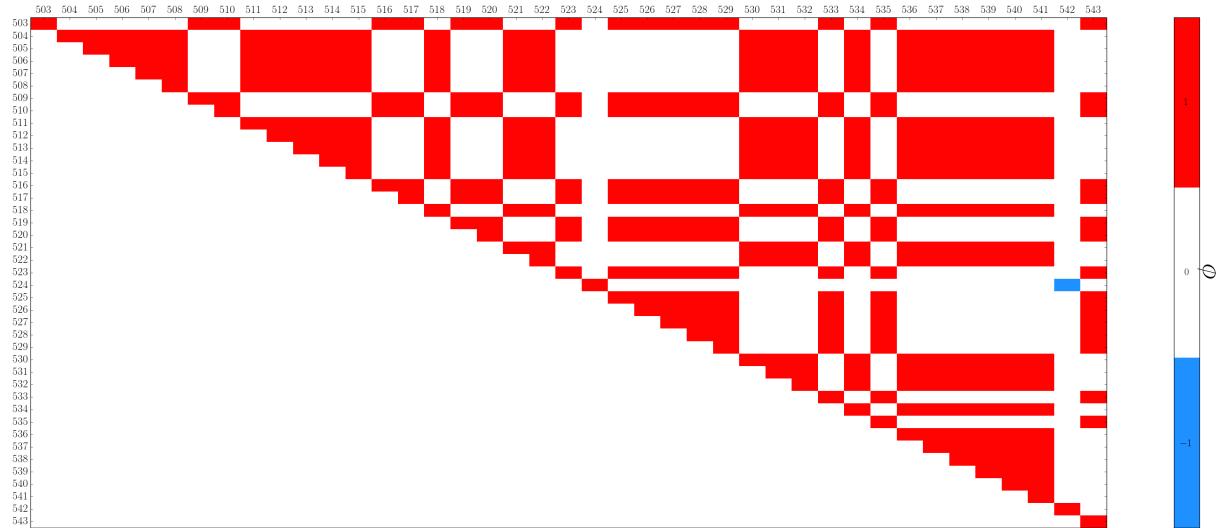


Figure 32: Heat map of the ϕ coefficient for the same exemplary molecule as shown before (Figures 1, 28, 25). The map was calculated with 0.4 cutoff.

The ϕ coefficient takes the values between -1 and 1 . If the value is close to 0 the correlation is not reliable. In the heat map, all values larger than the `cutoff` will be marked red, all lower

than $-1*cutoff$ are marked blue. The rest will remain white. The level of `cutoff` is defined by the user.

The script produces a matrix that is both written into the text file and printed in the form of the heat map. Figure 32 presents the heat map of the ϕ coefficient for an exemplary RNA molecule and its 10 frame-long molecular dynamics. The `CORRELATIONS.py` script uses as an input: the MINT output `pairs_in_time.csv` type, the cutoff and list of the numbers of nucleotides you would like to compute this coefficient for.

To compute the ϕ coefficient for all nucleotides type:

```
python CORRELATIONS.py filename_pairs.csv 0.4
```

where `filename` is name of your csv file and `0.4` is an example of the `cutoff` for these calculations. To compute the phi coefficient only for specified nucleotides type:

```
python CORRELATIONS.py filename_pairs.csv 0.4 "[ (1210,1220) , (985,995) ]"
```

The list of nucleotides has to be specified in square brackets and quotation marks. Inside the round brackets the ranges of nucleotides must be specified.

The heat map is only printed for the half of the matrix because the matrix is symmetric and the other half is identical. All nucleotides are correlated with themselves so the diagonal is red. All nucleotides that form the WC pairs with each other are also positively correlated. The neighbouring pairs that work together will be also correlated in all to all manner. The negative correlation suggests that there are pairs that open while others close.

9 Appendices

.1 Adding hydrogens to a PDB structure

The hydrogen bond definition used in MINT requires hydrogen atoms in the structure in order to measure the angle. Therefore, the input file has to contain all atoms.

If one analyzes the trajectory files derived from MD simulations, the hydrogen atoms were added to the structure during the preparation of an MD run, according to the atom type definitions in the used force field. Various tools from the MD packages can be used to assign the positions of hydrogens (for example, Xleap from AmberTools).

However, if a structure from the PDB database has to be analyzed and the user does not have experience with the MD methods, hydrogen atoms can be added using on-line servers. We have tested a few and recommend the Molprobity service [6] (<http://molprobity.biochem.duke.edu/>). It works even with the structures as large as the ribosomal subunits in an acceptable time span. What is more, the software can be downloaded from the <http://kinemage.biochem.duke.edu/software/reduce.php> website and used offline. The MINT webserver uses reduce program [25] to add hydrogenes to the structures downloaded from PDB.

.2 Exemplary description file

A fragment of the `_description` file generated during the analysis of an `example.dcd` trajectory.

Running with parameters:

```

MINT_home = /work/mint
Mode=Traj
OP_stacking_distance_cutoff=4.5
chains_names=[]
create_csvs=1
cutoff=20.0
cutoff_stacking=10.5
file_dcd=./example.dcd
file_name=./example.pdb
first_frame=0
force_field=AMBER
h_bond_angle=140.0
h_bond_atom=donor
h_bond_l=4.0
last_frame=-1
list_of_modified_nucs=data/unknown_modified.fa
margin=0.8
max_memory_GB=1.5
only_analysis=0
out_name=./traj-out/traj
stride=1
table_charges=/data/charges_and_VDW_modified.csv
table_nucleotides=/data/nucleotides_modified.csv
threads=2
time_cutoff=0.02
vdw_cutoff_stacking=-0.5
vmd=0

```

46 nucleotides, sequence:

```

GUA CYT ADE CYT CYT GUA GUA CYT URA ADE ADE CYT URA CYT CYT GUA URA GUA CYT CYT ADE GUA CYT ADE GUA CYT CYT GUA

```

Nucleotides statistics:

```

GUA-> 16,      parameters: known
ADE-> 9,       parameters: known
URA-> 6,       parameters: known
CYT-> 15,      parameters: known

```

Average secondary structure :

```

GCACCGGCUAACUCGGUGGCCAGCAGCCGCGGUAAUACGGAGGGUGC
((((.....((((.....((....)).....)))))))

```

frame number 0

Helices:

- 1] N|GUA:500-N|CYT:504 -> N|GUA:541-N|CYT:545
- 2] N|CYT:511-N|GUA:515 -> N|CYT:536-N|GUA:540
- 3] N|GUA:521-N|CYT:522 -> N|GUA:527-N|CYT:528

Helices vmd:

- 1] chain N and resid 500 to 504 chain N and resid 541 to 545
- 2] chain N and resid 511 to 515 chain N and resid 536 to 540
- 3] chain N and resid 521 to 522 chain N and resid 527 to 528

Pseudoknots:

- 1] N|GUA:505-N|CYT:526
- 2] N|GUA:506-N|CYT:525
- 3] N|CYT:507-N|GUA:524

Pseudoknots vmd:

- 1] chain N and resid 505 526
- 2] chain N and resid 506 525
- 3] chain N and resid 507 524

Motifs:

- 1] 0-6 N|CYT:504-N|GUA:541-N|GUA:540-N|CYT:511-N|ADE:510-N|ADE:509-N|URA:508-N|CYT:507-N|GUA:506-N|GUA:505-N
- 2] 7-5 N|GUA:515-N|CYT:536-N|ADE:535-N|URA:534-N|ADE:533-N|ADE:532-N|URA:531-N|GUA:530-N|GUA:529-N|CYT:528-N
- 3] 4 N|CYT:522-N|GUA:527-N|CYT:526-N|CYT:525-N|GUA:524-N|ADE:523-N|CYT:522-

Motifs vmd:

- 1] chain N and resid 504 541 540 511 510 509 508 507 506 505 504
- 2] chain N and resid 515 536 535 534 533 532 531 530 529 528 521 520 519 518 517 516 515
- 3] chain N and resid 522 527 526 525 524 523 522

WC Base pairs:

- 0] N|GUA:500/N|CYT:545 N2-H22-O2 angle: 163.09 distance: 2.94 | N1-H1-N3 angle: 171.83 distance: 3.32
| O6-H42-N4 angle: 163.34 distance: 3.51 WC/WC/3 Cis
- 1] N|CYT:501/N|GUA:544 N4-H42-O6 angle: 152.54 distance: 3.57 | O2-H21-N2 angle: 152.72 distance: 2.64
| O2-H1-N1 angle: 141.73 distance: 3.03 | N3-H1-N1 angle: 154.53 distance: 3.12 WC/WC/4 Cis
- 2] N|ADE:502/N|URA:543 N6-H61-O4 angle: 169.23 distance: 2.72 | N1-H3-N3 angle: 162.94 distance: 3.05 WC/WC/
Cis
- 3] N|CYT:503/N|GUA:542 N4-H41-O6 angle: 176.06 distance: 3.12 | O2-H21-N2 angle: 174.83 distance: 2.89
| N3-H1-N1 angle: 179.08 distance: 2.97 WC/WC/3 Cis
- 4] N|CYT:504/N|GUA:541 N4-H41-O6 angle: 152.44 distance: 3.46 | O2-H21-N2 angle: 161.5 distance: 2.77
| N3-H1-N1 angle: 159.49 distance: 3.17 WC/WC/3 Cis
- 5] N|GUA:505/N|CYT:526 N2-H21-O2 angle: 172.83 distance: 2.9 | N1-H1-N3 angle: 159.09 distance: 2.82 WC/WC/
Cis
- 6] N|GUA:506/N|CYT:525 N2-H21-O2 angle: 163.89 distance: 2.89 | N1-H1-N3 angle: 165.34 distance: 3.02 WC/WC/
Cis
- 7] N|CYT:507/N|GUA:524 N4-H41-O6 angle: 150.96 distance: 2.78 | O2-H21-N2 angle: 164.61 distance: 2.87
| N3-H1-N1 angle: 152.92 distance: 2.85 WC/WC/3 Cis
- 8] N|CYT:511/N|GUA:540 N4-H41-O6 angle: 150.02 distance: 3.04 | O2-H21-N2 angle: 158.9 distance: 2.86
| O2-H1-N1 angle: 140.91 distance: 3.25 | N3-H1-N1 angle: 157.45 distance: 3.01 WC/WC/4 Cis
- 9] N|URA:512/N|ADE:539 N3-H3-N1 angle: 148.45 distance: 3.07 | O4-H62-N6 angle: 160.03 distance: 3.52 WC/WC/
Cis
- 10] N|CYT:513/N|GUA:538 N4-H42-O6 angle: 159.98 distance: 3.08 | O2-H21-N2 angle: 158.07 distance: 3.06
| N3-H1-N1 angle: 167.56 distance: 3.17 WC/WC/3 Cis
- 11] N|CYT:514/N|GUA:537 N4-H41-O6 angle: 165.28 distance: 2.98 | O2-H21-N2 angle: 159.94 distance: 2.96
| N3-H1-N1 angle: 163.17 distance: 2.92 WC/WC/3 Cis
- 12] N|GUA:515/N|CYT:536 N2-H22-O2 angle: 169.37 distance: 2.75 | N1-H1-N3 angle: 176.86 distance: 2.84
| O6-H41-N4 angle: 152.03 distance: 2.94 WC/WC/3 Cis
- 13] N|GUA:521/N|CYT:528 N2-H21-O2 angle: 153.57 distance: 2.92 | N1-H1-N3 angle: 167.85 distance: 3.02
| O6-H42-N4 angle: 168.03 distance: 2.82 WC/WC/3 Cis
- 14] N|CYT:522/N|GUA:527 N4-H42-O6 angle: 154.51 distance: 2.95 | O2-H21-N2 angle: 168.62 distance: 2.71
| N3-H1-N1 angle: 164.77 distance: 3.01 WC/WC/3 Cis

non-WC Base pairs:

- 0] N|CYT:503/N|ADE:510 O2'-H2'-N1 angle: 156.21 distance: 2.89 Sugar/WC/1 Trans
- 1] N|GUA:505/N|CYT:525 O6-H41-N4 angle: 141.08 distance: 3.3 WC*Hoogsteen/WC*Hoogsteen/1 Cis
- 2] N|ADE:509/N|URA:543 O2'-H2'-O2' angle: 154.91 distance: 2.76 | N3-H2'-C2' angle: 156.62 distance: 3.81
| N3-H2'-O2' angle: 153.66 distance: 2.88 Sugar/Sugar/3 Cis
- 3] N|ADE:510/N|GUA:542 O2'-H2'-N3 angle: 154.88 distance: 3.03 Sugar/Sugar/1 Cis
- 4] N|URA:516/N|CYT:519 C2'-H2'-N3 angle: 141.12 distance: 3.8 Sugar/WC/1 Cis
- 5] N|URA:516/N|ADE:533 N3-H3-N7 angle: 169.81 distance: 2.92 | O2-H62-N6 angle: 159.53 distance: 2.89 WC/Hoo
Trans
- 6] N|GUA:517/N|GUA:529 N3-H2'-O2' angle: 148.04 distance: 3.27 Sugar/Sugar/1 Trans
- 7] N|CYT:519/N|GUA:529 N4-H42-O2' angle: 161.25 distance: 3.04 WC*Hoogsteen/Sugar/1 Trans
- 8] N|ADE:520/N|GUA:529 N6-H62-N3 angle: 172.32 distance: 2.98 | N7-H22-N2 angle: 143.75 distance: 3.1 Hoogste
Cis
- 9] N|ADE:520/N|ADE:533 N1-H61-N6 angle: 154.34 distance: 3.27 WC/WC*Hoogsteen/1 Trans
- 10] N|GUA:521/N|CYT:536 N3-H2'-O2' angle: 162.37 distance: 3.2 Sugar/Sugar/1 Cis
- 11] N|ADE:523/N|CYT:525 O2'-H42-N4 angle: 155.11 distance: 2.95 Sugar/WC*Hoogsteen/1 Trans
- 12] N|ADE:523/N|CYT:526 N3-H42-N4 angle: 159.81 distance: 3.18 Sugar/WC*Hoogsteen/1 Trans
- 13] N|GUA:527/N|ADE:535 N2-H22-N3 angle: 152.12 distance: 3.63 | O2'-H2'-N1 angle: 151.27 distance: 2.69 Sug
Trans
- 14] N|CYT:528/N|ADE:535 C2'-H2'-O2' angle: 151.97 distance: 3.98 | O2'-H2'-O2' angle: 161.7 distance: 3.11 S
Trans
- 15] N|ADE:533/N|ADE:535 C2-H2-O2' angle: 144.55 distance: 3.34 Sugar/Sugar/1 Trans

Dot-Bracket

GCACCGGCUAACUCGGUGGCCAGCAGCCGGGUAAUACGGAGGGUGC

(((((.....((.....((....))))))))))

Modified nucleotides denoted by lower case letters

Stacking pairs : qn1 n2 Coul VDW sum

N|GUA:500 N|CYT:501(-6.29, -5.89, -12.18)
N|CYT:501 N|ADE:502(10.02, -3.16, 6.86)
N|CYT:501 N|CYT:545(5.24, -1.39, 3.85)
N|ADE:502 N|CYT:503(4.19, -6.52, -2.33)
N|ADE:502 N|GUA:544(0.51, -5.39, -4.88)
N|CYT:503 N|CYT:504(3.13, -3.4, -0.27)
N|CYT:503 N|URA:543(3.71, -1.79, 1.92)
N|CYT:504 N|ADE:510(0.64, -0.8, -0.16)
N|CYT:504 N|CYT:511(-2.72, -2.05, -4.77)
N|CYT:504 N|GUA:542(1.24, -4.35, -3.1)
N|GUA:505 N|GUA:506(8.19, -6.41, 1.78)
N|GUA:505 N|GUA:527(-1.44, -0.54, -1.98)
N|GUA:505 N|ADE:535(1.47, -7.22, -5.75)
N|GUA:506 N|CYT:507(-3.71, -5.85, -9.57)
N|GUA:506 N|CYT:526(1.53, -2.79, -1.26)
N|CYT:507 N|URA:508(2.19, -4.53, -2.34)
N|CYT:507 N|CYT:525(3.06, -0.82, 2.24)
N|ADE:509 N|ADE:510(22.56, -5.18, 17.38)
N|ADE:509 N|GUA:544(2.16, -0.51, 1.65)
N|ADE:510 N|URA:543(3.98, -0.73, 3.25)
N|CYT:511 N|URA:512(1.98, -5.21, -3.23)
N|CYT:511 N|GUA:541(3.77, -3.82, -0.06)
N|URA:512 N|CYT:513(1.02, -3.99, -2.97)
N|URA:512 N|GUA:540(-1.38, -2.04, -3.42)
N|CYT:513 N|CYT:514(4.8, -3.89, 0.91)
N|CYT:513 N|ADE:539(5.91, -2.01, 3.9)
N|CYT:514 N|GUA:515(-1.01, -3.71, -4.72)
N|CYT:514 N|GUA:538(-1.86, -3.66, -5.52)
N|GUA:515 N|URA:516(1.27, -6.98, -5.71)
N|GUA:515 N|GUA:537(0.37, -7.68, -7.31)
N|URA:516 N|GUA:517(2.82, -4.41, -1.59)
N|URA:516 N|ADE:520(5.69, -0.52, 5.17)
N|URA:516 N|CYT:536(3.5, -0.77, 2.73)
N|GUA:517 N|CYT:519(-0.37, -0.75, -1.12)
N|GUA:517 N|URA:531(3.51, -2.24, 1.27)
N|GUA:517 N|ADE:533(3.15, -2.41, 0.74)
N|CYT:518 N|GUA:529(3.87, -3.65, 0.22)
N|CYT:518 N|GUA:530(4.8, -5.94, -1.14)
N|CYT:519 N|ADE:520(0.29, -6.18, -5.89)
N|ADE:520 N|GUA:521(5.56, -5.03, 0.53)
N|ADE:520 N|CYT:536(5.5, -1.1, 4.4)
N|GUA:521 N|CYT:522(-3.96, -5.2, -9.16)
N|GUA:521 N|GUA:529(-1.09, -4.27, -5.36)
N|CYT:522 N|ADE:523(-2.27, -3.71, -5.98)
N|CYT:522 N|CYT:528(8.7, -2.06, 6.64)
N|ADE:523 N|GUA:527(9.83, -5.65, 4.19)
N|GUA:524 N|CYT:525(-6.82, -4.95, -11.77)
N|CYT:525 N|CYT:526(3.63, -4.58, -0.95)
N|CYT:526 N|GUA:527(-0.2, -2.58, -2.79)
N|GUA:527 N|CYT:528(-5.27, -6.24, -11.51)
N|CYT:528 N|GUA:529(-1.71, -3.73, -5.45)
N|ADE:533 N|CYT:536(5.71, -4.87, 0.83)
N|CYT:536 N|GUA:537(4.54, -2.54, 2.0)
N|GUA:537 N|GUA:538(7.69, -6.53, 1.16)
N|GUA:538 N|ADE:539(9.86, -7.76, 2.11)
N|ADE:539 N|GUA:540(5.33, -5.77, -0.44)
N|ADE:539 N|GUA:541(2.77, -0.56, 2.21)
N|GUA:540 N|GUA:541(4.78, -6.8, -2.02)
N|GUA:541 N|GUA:542(6.33, -6.82, -0.5)
N|GUA:542 N|URA:543(0.42, -5.77, -5.34)
N|URA:543 N|GUA:544(5.16, -4.69, 0.47)
N|GUA:544 N|CYT:545(-7.19, -6.21, -13.4)

Stacking pi pairs

frame number 1

References

- [1] Raviprasad Aduri, Brian T. Psciuk, Pirro Saro, Hariprakash Taniga, H. Bernhard Schlegel, and John SantaLucia. Amber force field parameters for the naturally occurring modified nucleosides in rna. *Journal of Chemical Theory and Computation*, 3(4):1464–1475, 2007.
- [2] Pascal Auffinger. Personal communication, 2013.
- [3] Stacked Base, Dimers Comparison, and H-bonded Base. Nature of Nucleic Acid - Base Stacking : Nonempirical ab Initio and Empirical Potential. *Journal of Physical Chemistry*, 100:5590–5596, 1996.
- [4] Victor A. Bloomfield, Donald M. Crothers, and Ignacio Jr. Tinoco. *Nucleic Acids: Structures, Properties, and Functions*. University Science Books, 1999.
- [5] a P Carter, W M Clemons, D E Brodersen, R J Morgan-Warren, B T Wimberly, and V Ramakrishnan. Functional insights from the structure of the 30S ribosomal subunit and its interactions with antibiotics. *Nature*, 407(6802):340–8, September 2000.
- [6] Vincent B Chen, W Bryan Arendall, Jeffrey J Headd, Daniel a Keedy, Robert M Immormino, Gary J Kapral, Laura W Murray, Jane S Richardson, and David C Richardson. MolProbity: all-atom structure validation for macromolecular crystallography. *Acta crystallographica. Section D, Biological crystallography*, 66(Pt 1):12–21, January 2010.
- [7] Kevin Darty, Alain Denise, and Yann Ponty. VARNA: Interactive drawing and editing of the secondary structure. *Bioinformatics*, 25(15):1974–1975, 2009.
- [8] Dirk Evers and Robert Giegerich. RNA Movies : visualizing RNA secondary structure spaces. *Bioinformatics (Oxford, England)*, 15(1):32–37, 1999.
- [9] Nicolas Foloppe and Alexander D Mackerell. All-Atom Empirical Force Field for Nucleic Acids : I . Parameter Optimization Based on Small Molecule and Condensed Phase Macromolecular Target Data. *Journal of Computational Chemistry*, 21(2):86–104, 2000.
- [10] Fonseca Guerra, F Matthias Bickelhaupt, Jaap G Snijders, De Boelelaan, NL-H V Amsterdam, and Theoretische Chemie. Hydrogen Bonding in DNA Base Pairs : Reconciliation of Theory and Experiment. *Journal of the American Chemical Society*, (122):4117–4128, 2000.
- [11] Pavel Hobza. Stacking interactions. *Physical Chemistry Chemical Physics*, 10(19):2595–2610, May 2008.
- [12] W Humphrey, A Dalke, and K Schulten. VMD: visual molecular dynamics. *Journal of Molecular Graphics*, 14(1):33–8, 27–8, February 1996.
- [13] Neocles B Leontis, Jesse Stombaugh, and Eric Westhof. The non-Watson-Crick base pairs and their associated isostericity matrices. *Nucleic acids research*, 30(16):3497–531, August 2002.
- [14] A Lescoute and Eric Westhof. The interaction networks of structured RNAs. *Nucleic Acids Research*, 34(22):6587–604, January 2006.
- [15] Jerzy Leszczynski, Pavel Hobza, and J Heyrovsky. S Electronic Properties , Hydrogen Bonding , Stacking , and Cation Binding of DNA and RNA Bases. *Biopolymers*, 61:3–31, 2002.
- [16] Alexander D Mackerell and Nilesh K Banavali. All-Atom Empirical Force Field for Nucleic Acids: II . Application to Molecular Dynamics Simulations of DNA and RNA in Solution. *Journal of Computational Chemistry*, 21(2):105–120, 2000.
- [17] Naveen Michaud-Agrawal, Elizabeth J. Denning, Thomas B. Woolf, and Oliver Beckstein. MDAnalysis: A Toolkit for the Analysis of Molecular Dyanmic Simulations. *Journal of Computational Chemistry*, 2012.
- [18] Eric F Pettersen, Thomas D Goddard, Conrad C Huang, Gregory S Couch, Daniel M Greenblatt, Elaine C Meng, and Thomas E Ferrin. UCSF Chimera—a visualization system for exploratory research and analysis. *Journal of computational chemistry*, 25(13):1605–12, October 2004.
- [19] James C Phillips, Rosemary Braun, Wei Wang, James Gumbart, Emad Tajkhorshid, Elizabeth Villa, Christophe Chipot, Robert D Skeel, Laxmikant Kalé, and Klaus Schulten. Scalable molecular dynamics with NAMD. *Journal of Computational Chemistry*, 26(16):1781–802, December 2005.
- [20] Everitt B. S. *The Analysis of Contingency Tables*. Chapman and Hall, second edition, 1977.
- [21] S. Smit, K. Rother, J. Heringa, and R. Knight. From knotted to nested RNA structures: a variety of computational methods for pseudoknot removal. *RNA*, 14(3):410–416, Mar 2008.

- [22] J Sponer, H a Gabb, J Leszczynski, and P Hobza. Base-base and deoxyribose-base stacking interactions in B-DNA and Z-DNA: a quantum-chemical study. *Biophysical journal*, 73(1):76–87, July 1997.
- [23] Jií Sponer, Jerzy Leszczynski, and Pavel Hobza. Base Stacking in Cytosine Dimer. A Comparison of Correlated Ab Initio Calculations with Three Empirical Potential Models and Density Functional Theory Calculations. *Journal of Computational Chemistry*, 17(7):841–850, 1995.
- [24] Junmei Wang, Piotr Cieplak, and Peter a. Kollman. How well does a restrained electrostatic potential (RESP) model perform in calculating conformational energies of organic and biological molecules? *Journal of Computational Chemistry*, 21(12):1049–1074, September 2000.
- [25] J M Word, S C Lovell, J S Richardson, and D C Richardson. Asparagine and glutamine: using hydrogen atom contacts in the choice of side-chain amide orientation. *Journal of molecular biology*, 285(4):1735–47, January 1999.